Vision-based Methods for Evaluating Visualizations

Von der Fakultät Informatik, Elektrotechnik und
Informationstechnik der Universität Stuttgart
zur Erlangung der Würde eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

Vorgelegt von

Rudolf Netzel

aus Kaptschagai (KAZ)

Hauptberichter:	Prof. Dr. Daniel Weiskopf
Mitberichter:	Prof. Dr. Andrew T. Duchowski
Tag der mündlichen Prüfung:	22. Juni 2018

Visualisierungsinstitut
der Universität Stuttgart

2018

Acknowledgments

First of all, I would like to thank Daniel Weiskopf, my supervisor, for giving me the opportunity to be his PhD student. He provided me the best support and supervision I can imagine. Thank you, Daniel, for all your time, wisdom, patience, and support! Without you, I would not have been a PhD student and performed all the interesting work I present in this dissertation. Furthermore, I would like to thank Andrew Duchowski for being reviewer for this dissertation and taking part in my PhD defense. I am very glad that Benjamin and Markus Höferlin supervised my student and Diplom thesis. This laid the foundation for my PhD and without this time, I don't think I would have started my PhD at VISUS. I also want to thank Thomas Ertl and Daniel Weiskopf for making the fantastic working environment at VIS/VISUS possible. I would also like to thank the people with whom I shared the offices during my time as a PhD student: Marco Ament, Marcel Hlawatsch, Hansjörg Schmauder, and Haris Mumtaz. Special thanks go to Kuno Kurzhals, Bastian Reitschuster, Grzegorz Karch, Christoph Schulz, Michael Ludwig, Michael Burch, Nils Rodrigues, and Marcel Hlawatsch for proof reading this thesis and their valuable comments. Furthermore, I thank Martin Falk for creating and providing the Latex template for the thesis. What makes the time so special at VIS/VISUS are the nice people and the different activities and the fun we have together. Therefore, I want to thank the following persons for being nice colleagues and their support during my work: Julian Heinrich, Fabian Beck, Markus Huber, Corinna Vehlow, Marco Ament, Michael Raschke, Tanja Blascheck, Sebastian Boblest, Tanja Munz, Moataz Abdelaal, Valentin Bruder, Oliver Fernandes, Steffen Frey, Michael Bußler, Markus John, Florian Heimerl, Qi Han, Florian Haag, Robert Krüger, David Körner, Anette Müller, Enrico Taras, Benjamin and Markus Höferlin, Thomas Müller, Wolfgang Bayerlein, Tina Barthelmes, Maria Schulz, Karin Vrana, Martin Schmid, Ulrike Ritzmann, Margot Roubicek, Sophie Schroth, and Christine Schütz. Finally, I want to express my deepest gratitude to my family, especially my parents, who supported me by letting me go my way and not putting any pressure on me, and also my grandparents how are no longer able to wideness my graduation. Finally, I thank all my friends, who keep my life in balance and give me back the energy I need for my work: Basti, Wolfgang, Katja, Alex, Desi, Reuse, Seirus, Cernd, Aiala, Ayumi, all of the Michaels, Leonard, Lorenz, Daniel, Anne, Dagmar, and Peter. My work at the Visualization Research Center of the University of Stuttgart was funded by the German Research Foundation DFG within the SFB/Transregio 161.

Rudolf Netzel

Contents

List of Figures

Chapter 1

Chapter 2

Part I

Chapter 3

Chapter 4

Part II

Chapter 5

Chapter 6

Chapter 7

Chapter 8

List of Tables

Chapter 3

Chapter 5

Chapter 6

Chapter 7

Chapter 8

List of Abbreviations and Acronyms

CPU	central processing unit
GPU	graphics processing unit
RAM	random access memory
LGN	lateral geniculate nucleus
EEG	electroencephalography
EOG	electrooculography
RPSD	radial power spectrum diagram
BFECC	back and forth error compensation and correction
VOG	video-oculography
SSQ	sum of squares
MSSQ	mean sum of squares
DoF	degrees of freedom
AOI	area of interest
LIC	line integral convolution
OLIC	oriented line integral convolution
DSSM	dimensionally stacked similarity matrix
AMT	Amazon Mechanical Turk
GPS	global positioning system
SFC	space-filling curve
H	hypothesis
N	no direction encoding
A	standard arrow
EA	equidistant arrows
C	equidistant comets
T	tapered links
RND	random edge sorting
SF	shorter links on top of longer links
LF	longer links on top of shorter links
SP	splats for point visualization of nodes
NSP	no splats for point visualization of nodes
D1	low data density
D2	high data density
WA	within-image annotation
GA	grid reference annotation

DA	directional annotation
MA	miniature annotation
U	miniature annotation
U0, U1, U2	number of minimum needed transfers
C1, C2, C3	low, medium, and high map complexity
J	jump
NJ	nothing/jump
JD	jumping in direction
JF	jump/follow
F	follow
T	transfer
FTR	follow/transfer
FTU	follow/turn
N	nothing
S	start
E	end
FILA	start or end of an eye-tracking trajectory
CC	Cartesian coordinates
PC	parallel coordinates

List of Symbols

Symbol	**Explanation**
z	z-distribution
t	t-distribution
N	total number of observations
n_i	number of values in distribution i
F	F-value
p	p-value
t	time
ϕ	scalar quantity
$\frac{\partial \phi}{\partial t}$	partial derivative of a scalar quantity with respect to time
v	velocity
D	diffusion coefficient
e_i	error of advection
f_i	sources of errors
σ	standard deviation
sv_{i,j,M_k}	similarity value between participant P_i and P_j using metric M_k
M_{k,P_i}	value of metric k for participant P_i
d_d	distance deviation
d_c	distance to point C
d_b	distance to point B
H	equivalent for an F-value in parametric tests
r_{ij}	rank of values j in distribution i
$\bar{r}_i$	average rank of all values in distribution i
$\bar{r}$	the average of all r_{ij}
g	number of distributions
U_i	key value for measuring the overlap of distribution i with other distributions
R_i	sum of all ranks of distribution
χ^2	chi-square distribution
CI	confidence interval
$E(X)$	expected value of distribution X

Abstract

The analysis of data is an important task of a variety of domains, e.g., in the automotive industry to improve the design of cars and engines, economy to make prognoses based on current sales records, or medicine to measure the effects of treatments. With technological progress, the amount of data that needs to be processed has increased to a level that humans are no longer capable of handling manually. Therefore, a variety of visualization frameworks, tools, and models have been introduced to support humans during the analysis of large amounts of data.

Since analysts, or in general observers, deal with visual representations of data, human vision is essential and should be considered in the analysis or evaluation of data. To this end, this thesis contributes vision-based methods for the analysis or evaluation of data. The term vision-based refers here to either the utilization of perceptual models or the derivation of information of eye movements that are recorded using eye tracking.

The first part of this thesis focuses on the analysis of data. Here, models of human perception are used for quantification or compensation of visual effects. In the field of flow visualization, the blurring of an advected texture, introduced by the diffusion during the advection processes, is quantified based on a frequency model. Furthermore, models for perceived contrast, direction, and frequency are used to compensate for illusionary tilt effects of superimposed textures. Besides such automatic data analysis methods, the first part of this thesis also addresses frameworks for the user-driven analysis of eye-tracking data. Here, the main task of analysts is to identify behavioral patterns. Therefore, frameworks are presented that support analysts in three areas that are important in eye tracking: annotation of eye-gaze data, spatiotemporal data analysis, and metrics-based analysis.

The second part of this thesis also targets to find patterns in recorded eye-gaze data, but of recorded eye-tracking data is utilized to quantify task performance and to verify the existence of behavioral patterns. To this end, eye-tracking user studies are conducted or evaluated in four domains: trajectory visualization, correspondence visualization, automatic extraction of groups of participants with similar eye-movement patterns using maps, and the comparison of Cartesian and parallel coordinates. The evaluation scheme for each study is designed according to the used tasks and stimuli. Based on the results of studies, recommendations or guidelines for the usage of the tested visualization are derived.

German Abstract
—Zusammenfassung—

Die Analyse von Daten ist in vielen Bereichen eine wichtige Aufgabe, z. B. in der Automobilindustrie, um das Design von Autos und Motoren zu verbessern, in der Wirtschaft, um Prognosen basierend auf aktuellen Verkaufszahlen zu erstellen, oder in der Medizin, um die Auswirkungen von Behandlungen zu messen. Mit dem technologischen Fortschritt hat sich die Menge an Daten, die verarbeitet werden muss, auf ein Niveau erhöht, das Menschen nicht mehr manuell bewältigen können.

Da sich Analysten oder allgemein Beobachter mit visuellen Darstellungen von Daten beschäftigen, ist das menschliche Sehen wichtig und sollte bei der Analyse oder Auswertung von Daten berücksichtigt werden. Diesbezüglich werden in dieser Arbeit Sicht-basierte Methoden zur Analyse oder Auswertung von Daten diskutiert. Der Begriff Sicht-basiert bezieht sich hier entweder auf die Verwendung von Wahrnehmungsmodellen oder der Ableitung von Informationen aus Augenbewegungen, die mit Eye-Tracking aufgezeichnet werden.

Der erste Teil dieser Arbeit konzentriert sich auf die Analyse von Daten. Hier werden Modelle der menschlichen Wahrnehmung zur Quantifizierung oder Kompensation von visuellen Effekten verwendet. In dem Gebiet der Strömungsvisualisierung wird das Verschwimmen von advektierten Texturen anhand eines Frequenzmodells quantifiziert, welches auf die Diffusion während des Advektionsvorgangs zurückzuführen ist. Darüber hinaus werden Modelle für wahrgenommenen Kontrast, Richtung und Frequenz verwendet, um illusionäre Neigungseffekte überlagerter Texturen zu kompensieren. Neben solchen automatischen Datenanalyseverfahren behandelt der erste Teil dieser Arbeit auch Rahmenwerke für die nutzergesteuerte Analyse von Eye-Tracking-Daten. Hier besteht die Hauptaufgabe von Analysten darin, Verhaltensmuster zu identifizieren. Daher werden Rahmenwerke vorgestellt, die Analysten in drei Bereichen unterstützen, die beim Eye-Tacking von Bedeutung sind: Annotation von Blickdaten, raumzeitliche Datenanalyse und metrikbasierte Analyse.

Der zweite Teil dieser Arbeit zielt auch darauf ab, Muster in aufgezeichneten Blickdaten zu finden, aber es werden aufgezeichnete Eye-Tracking-Daten verwendet, um die Aufgabenleistung zu quantifizieren und die Existenz von Verhaltensmustern zu verifizieren. Zu diesem Zweck werden Eye-Tracking-Benutzerstudien in vier Bereichen durchgeführt oder evaluiert: Trajektorienvisualisierung, Korrespondenzvisualisierung, automatische Extraktion von Gruppen von Teilnehmern mit ähnlichen Augenbewegungsmustern unter Verwendung von Karten und der Vergleich von kartesischen und parallelen Koordinaten. Das Evaluationsschema für jede Studie wurde unter Berücksichtigung der verwendeten Aufgaben und Stimuli entworfen. Basierend auf den Ergebnissen von Studien werden Empfehlungen oder Richtlinien für die Verwendung der getesteten Visualisierungen abgeleitet.

Chapter 1

Introduction

In general, vision is one of the most important human senses that we use effortlessly despite the high complexity of the human visual system. Although it is highly complex, scientists were able and are still in the process of understanding the human visual system by generating models that approximate functions of the visual system. In some domains, such models are of importance, e.g., in computer science for the implementation of visualization methods. Here, visualizations can be improved by taking into account models of human vision. Furthermore, they can be used for the quantification of visual quality or visual effects.

Since visualization is becoming increasingly important, there is a need for further quantification. To address the issue of quantification in visualization and to foster collaborative research between universities, a large-scale research project was established and is being funded by the DFG (German Research Foundation): SFB/ Transregio 161 ("Quantitative Methods for Visual Computing").

In this thesis, the results of my research at the University of Stuttgart and within project B01 ("Adaptive Self-Consistent Visualization") of the SFB/ Transregio 161 are presented. The focus of my work was to gain insights into visual behavioral patterns and cognitive processes of observers by employing eye-gaze data and data derived from it. The eye-gaze data was recorded by use of an eye-tracking device. To this end, methods for a user-driven analysis of such eye-tracking data were developed to support analysts during their analysis. As an alternative to such user-driven data analysis methods, eye-tracking user

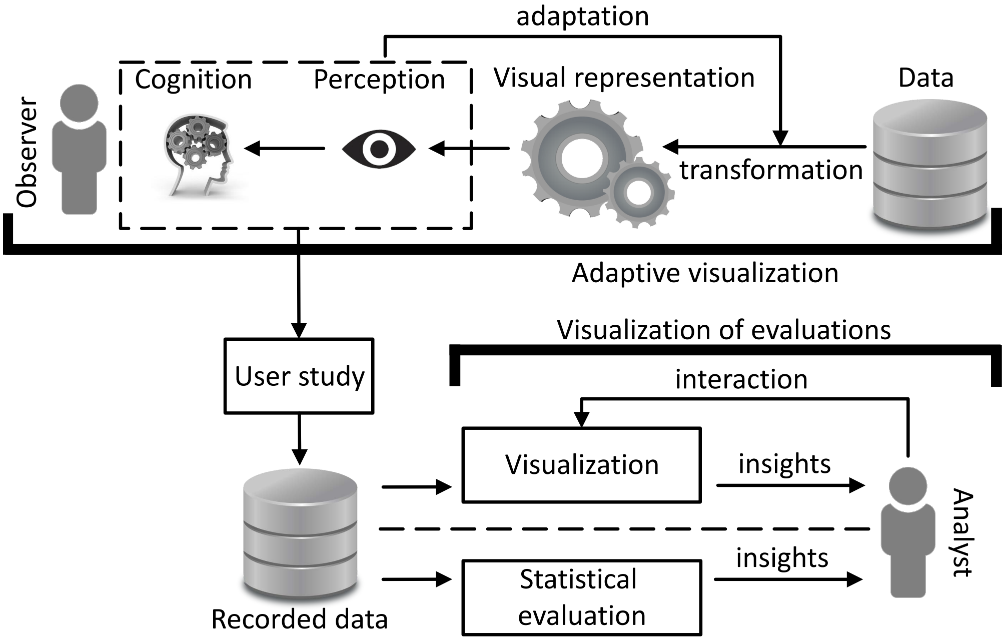


Figure 1.1 — Workflow and components involved in data analysis. A representation of the data is generated and an automatic analysis of the data based on perceptual models can be used for an adaptation of the representation. Alternatively, such representations can be subject of a user study, where data is recorded that is relevant for the study. This data can now be analyzed by using classical visualization approaches or by means of a statistical evaluation. In both cases, an analyst gains insights.

studies were conducted, and the recorded data was statistically evaluated to verify hypotheses concerning visual behavioral patterns, cognition, and task performance of subjects. Another area that is addressed in this thesis is the utilization of human perception models to quantify the quality of visualizations and to perform an adaptation to enhance the perception of them.

1.1 Data Analysis and Eye Tracking

Eye tracking is nowadays used in many domains, e.g., in the automotive industry, marketing, medical science, computer science, or for the development of websites. This is a result of the fast development and progress of eye-tracking devices, which vary in shape and size, and can, therefore, be utilized in different scenarios. In general, eye tracking can be used to examine how observers scan a presented visual stimulus, thereby, extracting, e.g., areas of attention to improve the presentation or the design of a product. In the case of medical science, it can be used as an indicator of some neurological illnesses. Metrics derived from the eye-gaze data are utilized to infer mental workload or act as an indicator of cognitive processes. Often, such metrics are used in combination with other measured data like electroencephalography (EEG).

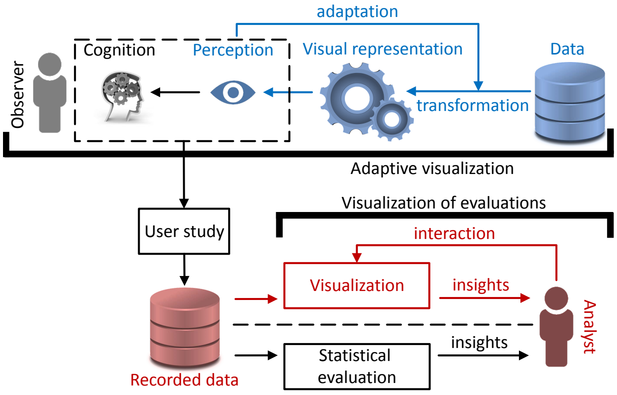
Depending on the domain and the data, there are different approaches to analyze data, but an overarching workflow can be derived that contains basic components. This is depicted in Figure 1.1. In the upper part of this workflow, an observer is processing the information of a visual representation of data. Here, the perception of the representation is essential and by using perceptual models the representation can be automatically adjusted to enhance it. The lower part contains different ways to handle the data. First, there is the visualization-based analysis of the data, where an analyst is interacting with a framework, while the second approach involves statistical testing. In either way, insights into the data are obtained by an analyst.

1.2 Research Question

The underlying research question of this thesis was to understand what influence visual representations have on observers and their eye movements. Therefore, it is necessary to provide data analysis and evaluation methods that incorporate human vision to identify behavior patterns or to improve the visual representations of data. Here, methods are required that utilize models for quantification of human perception. Eye tracking is another way to include human vision in the data analysis processes. Here, methods are necessary that support analysts during a data analysis of eye-tracking data, which include recorded eye-gaze data, derivatives of them, or other data obtained during eye tracking. An alternative to such user-driven data analysis approaches is to utilize eye-tracking data as basis for an empirical evaluation. Here, the first challenge is to design user studies, including tasks and stimuli, to confirm or reject hypotheses. The design of the evaluation schemes corresponds to the second challenge. They have to be adapted to the used tasks and stimuli to address established hypotheses appropriately. Figure 1.1 shows how the previously mentioned analysis and evaluation methods can be combined into one workflow for a human vision-based analysis and evaluation of either visual representations (images) or recorded eye-tracking data. In the following, methods related to the two identified major areas are discussed in more detail.

Methodology for Data Analysis (Part I) There are two basic approaches used in this thesis to perform data analysis. The first is based on models. The second approach is a user-driven analysis of data using visualization.

Models are used in general to describe complex circumstances on an algorithmic, a mathematical, or a procedural level. Especially when it comes to human perception, simplified models were established based on experiments. Such models include, e.g., color perception, edge detection, or adaption to different lighting conditions.



The main question is here, how such models can be used for quantification of visual quality of visualizations and how they can be utilized during the generation of the visualization or even after the generation, in a post-processing step. This corresponds to the part of the workflow highlighted in blue.

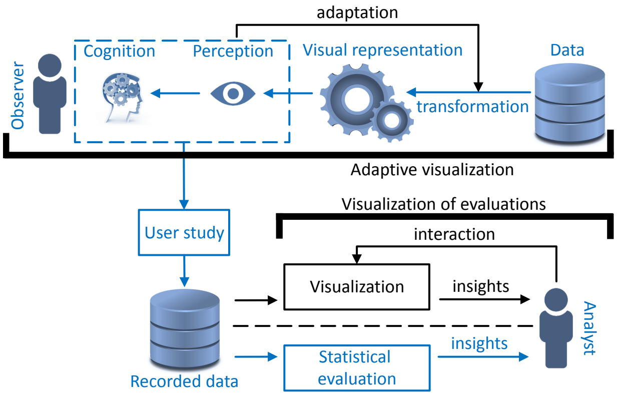
The purpose of visualization is to generate a truthful visual representation of data that needs to be analyzed. Unfortunately, neither all aspects of human perception can be considered during the generation of a visualization nor can all visual interaction effects be foreseen between elements in a visualization. Therefore, visualizations can contain parts that are misperceived by observers. Such effects could be detected and compensated by using perceptual models.

Another approach for the analysis of data is based on visualization or other applications that analysts can use to find patterns, interpret them, and, thereby, generate new insights. This corresponds to the part of the workflow highlighted in red. Usually, visualizations are utilized to interact with and represent data in a meaningful way depending on the scenario. They provide functions or interaction methods that should support an evaluation. Currently, there is no single best solution that can be used across all domains but rather a multitude of methods and visualizations that are domain or even task specific. This is also the case in the field of eye tracking. Here, the main question is how existing methods can be improved, or new ones created that support analysts in their data analysis.

There are three main topics in eye tracking that are addressed in this thesis: Metric selection, annotation, and (spatiotemporal) visualization of eye-tracking data. Although there are previous solutions for problems in these fields, there are still issues that need to be addressed.

Empirical Evaluation Using Eye Tracking (Part II) This part deals with a hypothesis-based evaluation of eye-tracking data as well as data that can be derived from eye movements.

Obtained measurements can be used for an inference of possible cognitive processes for a given task based on the perception of stimuli. All involved components of the workflow that are necessary to acquire, evaluate, or interpret data are highlighted in blue.



Eye tracking has a long history and is becoming more popular since the technological progress in this field has made it possible to use eye trackers in many domains. Therefore, a growing number of studies has been and, most likely, will be conducted to test how participants react to different stimuli or how given visualizations are scanned. Recorded eye-movement patterns are classically obtained by sequence analysis of areas of interest (AOIs) or by relying on metrics and hypotheses testing in addition to an evaluation of task performance by means of completion time and accuracy.

The main question of this part is if it is possible to go beyond the classical analysis methods by deriving new descriptive key values from eye-gaze data that are then used for hypothesis testing. By establishing such new ways of testing specific characteristics, the analysis toolbox will gradually be extended, and reported results would be more convincing, thereby, improving the quality and comparability of the presented research. Unfortunately, such derived values depend on the stimuli and the given tasks. Nevertheless, this thesis provides suggestions of new evaluation approaches for eye-gaze data for a variety of domains.

1.3 Outline and Contributions

This section provides a general outline of the publications presented in this thesis as well as the corresponding contributions. I am the first author of most of the presented publications, and I have performed all the evaluations as well as developed associated software or implemented necessary code. In the case that I was not the first author, I have written parts of the papers and was closely supervising the implementation of proposed methods. My supervisor Daniel Weiskopf was involved in all the work described here including discussions about it or proposing ideas, and contributing as a co-author to writing the respective publications.

This thesis is divided into two parts that address the previously described research questions. The first part deals with two kinds of methodology for data analysis: model-based evaluation with a focus on perception and application-based analysis methods for supporting analysts in understanding eye-tracking data.

In Chapter 3, two model-based approaches are presented that quantify aspects of human perception. The objective of the first approach was to understand the quality of texture advection in flow visualization better [1]. In particular, advection strategies, as well as their combination with interpolation methods of higher-order, are investigated. Radial power spectrum diagrams (RPSD) were proposed to compare different time steps of advection processes. Furthermore, a new method that combines higher-order interpolation was evaluated in addition to the influence of space discretization by testing different spatial resolutions of textures. The methods for advection were implemented by Stephan Starke in his diploma thesis [14] under the supervision of Marco Ament. I generated the used data and performed the evaluation. The paper was written in cooperation with my co-authors Marco Ament and Michael Burch, who helped with the writing and the proofreading. The second approach in Chapter 3 is not yet published and is based on a manuscript draft created by Nils Rodrigues and myself. Anja Haug implemented during her diploma thesis [15], under my supervision, a framework in Matlab to compensate the simultaneous orientation contrast. This contrast causes a tilt bias in the human perception if textures overlap that exhibit different perceived main directions.

Frameworks for the evaluations of eye-tracking data are described in Chapter 4 that support analysts in analyzing data concerning visible patterns. Three topics are addressed in this chapter: spatiotemporal visualization, annotation, and metric analysis. Concerning the spatiotemporal visualization of eye-tracking data, I implemented a method that reduces the 2D spatial domain to 1D, which allows one to use one dimension for encoding of temporal information [2]. The visualization combines traditional attention maps and temporal plots. The next topic addresses the issue of data annotation and problems that could occur with a traditional automatic AOI-based labeling. To this end, I proposed and implemented a lightweight annotation framework that facilitates efficient manual annotation [3]. It allows an annotator to work through a sequence of fixations, while it visualizes the context of the scanpath in the form of previous and subsequent fixations (sub-scanpath), while also providing context information of the stimulus. Static stimuli (images) and dynamic stimuli (videos) are supported. The paper was written together with my co-author Michael

Burch. The last topic discussed in Chapter 4 is a metric-based evaluation and the investigation of used metrics for eye-tracking data [4, 5]. Here, participants can be grouped based on dimensionally stacked similarity matrices. Furthermore, a method for the selection of metrics and generation of meta-metrics was proposed. Ayush Kumar was responsible for the implementation, which I was overlooking. Intermediate results were discussed together with Michael Burch. The paper was written by Ayush Kumar, Michael Burch, and myself, while Klaus Mueller was proofreading the paper.

In the second part of this thesis, different studies for an empirical evaluation of eye-tracking data are presented, for a variety of domains. In Chapter 5, the encoding of trajectories is evaluated [6]. Here, the goal was to test the efficiency of different directional encodings as well as node splatting used for the density estimation of nodes. The paper was written together with Michel Burch. I was conducting the study as well as performing the evaluation. This included necessary implementations. In Chapter 6, the effects of different word-sized visualizations were evaluated that can be used to establish a correspondence between a given label and its position on a 2D map [7]. A model was derived about how subjects solved a given task, which was shown using hypothesis testing. The execution of the study as well as the generation of the stimuli was performed by Sanjeev Balakrishnan as part of his Bachelor thesis [16] that was supervised by Hansjörg Schmauder and myself. I performed the evaluation of the eye-tracking data. The resulting paper was written in cooperation with Marcel Hlawatsch and Michael Burch. Chapter 7 reports the results of an experiment that aimed to evaluate metro maps and how subjects used them [8]. The study was statistically evaluated for different metrics. Furthermore, the eye movements were annotated and also statistically evaluated. The study was conducted by Bettina Ohlhausen as part of her diploma thesis [17] that was supervised by Michael Burch. I was performing the evaluation of the recorded data including all necessary implementations. The paper was written together with Kuno Kurzhals, Michael Burch, and Robin Woods, who provided the metro maps that were used as stimuli. In Chapter 8, task performance and reading characteristics for scatterplots and parallel coordinates were investigated with a controlled eye-tracking study [9]. The paper was written in cooperation with Jenny Vuong and Julian Heinrich, while Ulrich Engelke and Sean O'Donoghue were proofreading the paper. I conducted the study as well as performed the statistical evaluations of the eye-tracking data.

Material from papers [2, 4, 6, 7] under copyright of IEEE is reused with kind permission of IEEE following the agreement for reuse in a dissertation or thesis. Material from paper [5] is licensed by the Journal of Eye Movement Research under the Creative Commons Attribution 4.0 International License

(CC BY 4.0), which permits to share, copy, redistribute, adapt, remix, and transform the material in any medium or format. Material from paper [3] under the copyright of ACM is reused with kind permission of ACM following the agreement for reuse in a dissertation or thesis. Material from paper [1] under the copyright of Eurographics is reused with kind permission of Eurographics following the agreement for reuse in a dissertation or thesis. Material from paper [8] under the copyright of Taylor&Francis is reused with kind permission of Taylor&Francis following the agreement for reuse in a dissertation or thesis. Material from paper [9] is published under the Creative Commons Attribution-NonCommercial-NoDerivs license. Therefore, it is permanently free for everyone to read, download, copy, and distribute.

During my PhD research, I was involved in the publication of further papers that are not part of this thesis. Together with Beck et al. [10], I worked on word-sized visualizations for eye-tracking data. Another paper dealt with challenges of designing metro maps [11]. Concerning maps, I was working together with Rodrigues et al. [12] on a concept to visualize time series data with spatial context on the basis of energy production data of power plants. I was also involved in visualizing the training process of a classifier for online learning [13].

1.4 Hardware

All results and tool in this thesis were produced or implemented on the following system:

- Operating system: Windows 7 Professional, 64-bit
- CPU: Intel i7-2600 with 3.40 GHz
- RAM: 16 GB DDR 3
- GPU: nVidia Geforce GTX 560 Ti with 1024 MB GDDR5 video memory

All implementations for testing and evaluation are based on C++ and Qt using Visual Studio 2012, R in the version 3.03 in combination with RStudio, or Matlab 2016a. The conducted studies took place in a laboratory isolated from outside distractions. The room was artificially illuminated with dimmed lights. The laboratory space contained no distracting objects. The studies were conducted with a Windows PC driving a TFT screen with a resolution of 1920 × 1200 pixels. Eye-tracking data was recorded by a Tobii T60 XL eye-tracking system integrated into the TFT screen. The viewing distance was about 60 cm from the screen to allow for a good calibration of the eye-tracking system. Key parameters for the analysis software of the eye tracker were set to a minimum of 10 pixels covering and a minimum of 30 ms fixation duration.

CHAPTER 2

Basics

In this chapter, basic concepts are described that are related to this thesis, beginning with the human visual system from the input of light signals (the eye) to their processing (the brain) in Section 2.1. The next part provides information about eye tracking (Section 2.2) including descriptions of filters and metrics for eye-tracking data. In any case, an evaluation of this data should be performed, which is often done employing statistics. Therefore, a brief description of non-parametric and parametric tests is presented in Section 2.3. Additionally, the concept of new statistics for statistical evaluation is described. Eye-tracking data are often analyzed using visualization. To this end, Section 2.5 provides a basic categorization of visualization and analysis methods for eye-tracking data as well as the necessity for such approaches.

2.1 Human Visual System

Humans rely heavily on the sense of vision [18, 19, 20, 21, 22] in their daily lives, which is used as a basis for interactions with the physical world. Here, the eyes are usually used for planning purposes fixating objects before the interaction begins [23]. The eyes are constantly in motion, scanning the environment to send data to the brain that interprets the data, which results in an adjustment of the eye gaze. The eye movement and gaze are often correlated with the focus of attention, but there is still a certain amount of uncertainty when it comes to the actual estimation of attention areas, since humans are capable of looking at

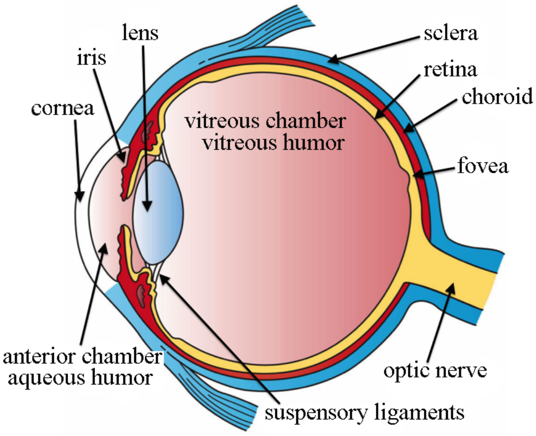


Figure 2.1 — Structural components of the human eye (Source: Wikimedia[1]).

one region in their field of view but actually paying attention to something in their peripheral field of vision [24].

Only by conducting experiments it is possible to understand the function of the visual system. Here, specialized stimuli are presented to a subject and the responses are measured. Obtained insights can be utilized for the generation of visualizations and also the design for other studies that rely on the visual system, which are both parts of this thesis.

The visual system can be decomposed into two functional parts: The eyes, which record and preprocess image data, and the brain, which further processes and interprets the data. Both units can be further decomposed into several sub-units.

The eye [25, 26] is a spherical object with a diameter of about 12 mm. The front part of the eye contains optical components that regulate the light income (pupil) and focus the light (lens deformed by muscles) on the retina in the back (see Figure 2.1). The retina is responsible for converting incoming light wave signals into electrical signals through chemical processes. Several layers in the retina perform this kind of low-level signal preprocessing. The different layers are depicted in Figure 2.2.

The first layer that processes light signals contains rods and cones. Here, rods are sensitive to brightness only and cones are sensitive to color. The distribution of such photoreceptors in the retina is shown in Figure 2.3. There are three type of cones, which respond to three different ranges of light wavelengths corresponding to L, M, and S. The fovea is the spot with the highest density of cones and, therefore, also the spot with the highest spatial resolution. The

---

[1] `https://commons.wikimedia.org/wiki/File:Three_Main_Layers_of_the_Eye.png`

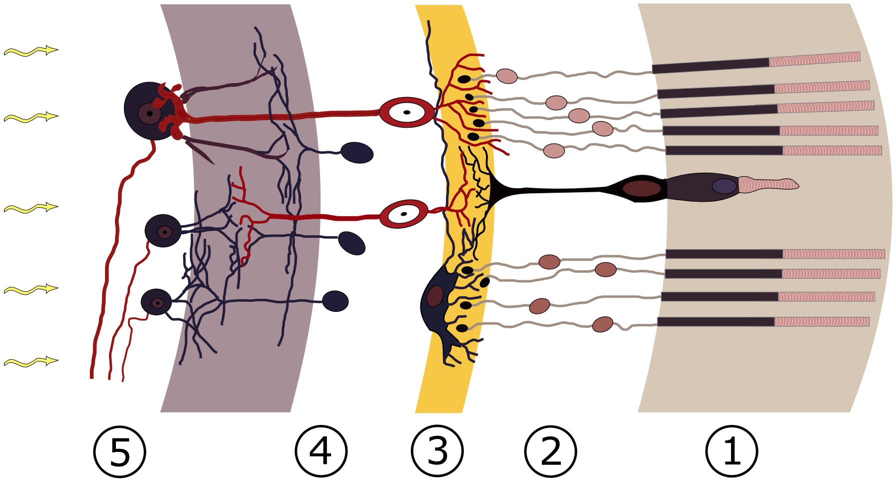


Figure 2.2 — Rods and cones photoreceptors ① generate luminance and color signals, which are aggregated to a certain degree by horizontal cells ②. Bipolar cells ③ transmit signals from the previous layers to the ganglion cells ⑤. The ganglion cells also receive input from amacrine cells ④, which aggregate signals from the bipolar cells. The ganglion cells transmit their output through the optical nerve to the brain. (Source: Wikimedia[2])

opposite is the case for the blind spot, where no photoreceptors are present at all since the optical nerve is located there. The signals generated by the first layer are then aggregated by horizontal cells in the second layer of the retina (see Figure 2.2) and are involved in the sensitivity adjustment of the rods and cones to bright and dark conditions. In dark environments, the horizontal cells are depolarized (low negative charge) due to the neurotransmitter level of the photoreceptors. This depolarization results in a hyperpolarization (higher negative charge) of the photoreceptors. In bright environments, the process is the opposite. The aggregation is based on receptive fields, and the output of a horizontal cell inhibits the signal of surrounding ones, which is also known as lateral inhibition.

In the third layer, bipolar cells transmit the signal from previous layers to successive layers. Similar to the horizontal cells, amacrine cells (layer 4) aggregate the signals from the bipolar cells.

The final layer consists of ganglion cells, which aggregate signals from previous layers. One result of this aggregation is the introduction of color-opponent signals by generating red-green, yellow-blue, and bright-dark signals.

The optical nerve is then used to transport the signals to the lateral geniculate nucleus (LGN), which is a relay station for them. Furthermore, the LGN

---

[2] `https://commons.wikimedia.org/wiki/File:Retina-diagram.svg`

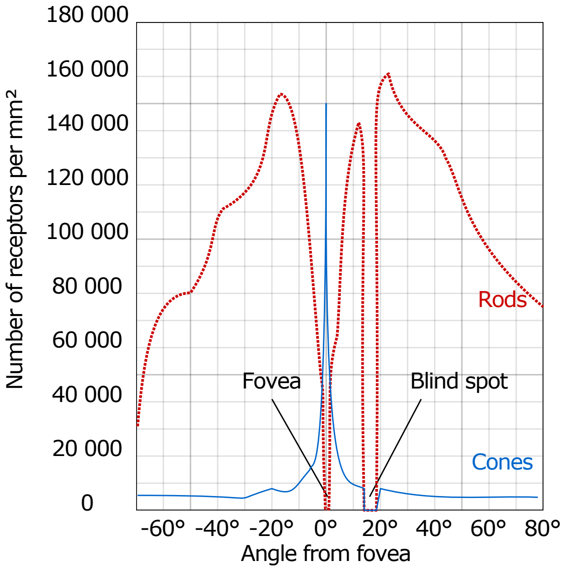


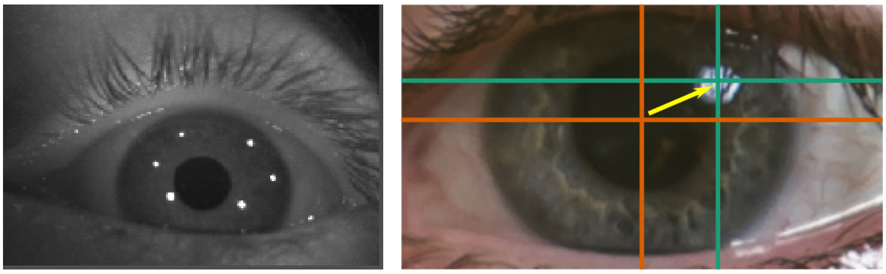
Figure 2.3 — Distribution of photoreceptors of rods (red) and cones (blue). The Highest spatial resolution is present at the fovea due to the highest density of cones, while the blind spot has no photoreceptors at all since the optical nerve is located there. (Source: Wikimedia[3])

performs non-mathematical calculations to generate temporal and spatially correlated/decorrelated data. Such calculations determine, e.g., major objects in space and their velocity.

The LGN forwards the signals to the visual cortex [27, 28]. Here, six layers are involved in processing the data. In all layers, receptive fields are tuned to different kinds of stimulation. The first layer (V1) can discriminate small changes in visual orientations, spatial frequencies, and colors. The determination of the spatial frequencies can be model by Gabor filters. The complexity of the tuning of the receptive fields is increasing in successive layers. This is shown in the response of the second layer (V2) that is additionally tuned to illusionary contours (and their orientation) and to determine background and foreground. Many things about third layer V3 are unclear, but it was shown that it is sensitive to orientation, color, and movement. Furthermore, V3 is assumed to play a role in the processing of global motion. In addition to the function of V2, layer four (V4) is tuned to intermediate object complexity (simple shapes). Until now, there are only spatial features extracted. In layer five (V5) temporal aspects are now considered by neurons that are tuned to speed and direction, e.g., by detecting corners or line endings. The last layer (V6) is dedicated to recognizing self-motion and low spatial frequency components.

---

[3] https://commons.wikimedia.org/wiki/File:Human_photoreceptor_distribution.svg



(a) (b)

Figure 2.4 — A human eye illuminated by (near-)infrared light (a) and under normal illumination (b). The bright spots indicate in both cases corneal reflections. The distance and direction from the center of the pupil are often used as features for gaze estimation as shown in (b).

2.2 Eye Tracking

The process of measuring either the position of gaze or the motion of the eyes is called eye tracking. Since the 1800s, it has been used to study eye movements. In the beginning, scientist relied on direct observations, and after the invention of the video camera, it was possible to record the observations and perform a subsequent analysis. In this context, Alfred L. Yarbus did important research in the field of eye tracking [29]. Through his experiments, he showed that the given task and stimulus primarily influence eye movements of subjects. Furthermore, he researched the relation between fixations and interest of subjects in the stimulus. To this end, this thesis presents experiments that use eye tracking to understand perception and cognition.

The basic concept of eye tracking did not change since then, but the used technologies improved and therefore the methodology was adjusted. Eye tracking nowadays means to extract features (e.g., pupil size, position, or orientation) of the eyes during the observation of their movements. These are then used for a gaze estimation on a 2D stimulus or even in a 3D scene. Computer vision algorithms extract features. They can be categorized in shape-based, appearance-based, or hybrid methods [30]. Shape-based methods are most commonly used. They rely on detecting the pupil, iris, or corneal reflections. These are then used to extract features like the distance and direction of a corneal reflection from the center of the pupil. An example of this is depicted in Figure 2.4. Appearance-based methods usually use machine learning to derive a model based on training data considering differing illumination and head poses. The actual gaze estimation is then done by using either extracted features, a trained model, or a combination of both in the case of a hybrid approach to get actual positions on a 2D stimulus or in a 3D scene. Approaches for gaze estimation can be categorized in geometry-based and interpolation-based methods. Methods belonging to the first category are based on more or less realistic models of the eyes and the scene. Based on the input data, the model reflects the position and orientation of the eyes. Assuming that the line of sight is a straight line that corresponds to the prolonged z-axis of the eye intersecting the center of the pupil, the gaze position can be calculated easily by intersecting the line of sight with the 2D stimulus or in the 3D scene. Interpolation-based methods often rely on a linear or quadratic functions that map gaze coordinates, expressed as features, to image coordinates. Within a calibration step, where subjects are asked to look at defined positions, features are recorded for each position, enabling the generation of an interpolation function that maps the feature space to the 2D or 3D space.

The features that are extracted depend on the used eye-tracking system. There are three different categories of systems [31]. The first one is based on electro-oculography (EOG) [32]. The basic idea behind EOG is to measure the potential of an electrical field created by the muscles of the eyes. It is a semi-invasive method since it requires placing sensors on the skin above and below the eyes as well as on the right and left side. The accuracy is not high, but the required equipment is not as expensive as other systems, and it can still be used for tasks where accuracy is not of high relevance. A highly accurate (10 arc seconds) but also more invasive method uses contact lenses [33]. To retrieve measurements, wires are connected to the lenses. Video-oculography (VOG) is the last category of eye-tracking systems, which rely on cameras that record the eyes. VOG is minimally invasive and also accurate with about 30 arc minutes depending on, e.g., the illumination and the properties of the hardware. Especially for VOG, there are three variations: remote, head-mounted, and mobile eye tracker.

A remote eye tracker is a system that contains the camera and the light source, and it is placed at a certain distance in front of a subject, e.g., it can be integrated into, or attached to, a screen. Subjects are free to move their head during an experiment, which affects the accuracy to a certain degree. Therefore, in some experimental setups, a chin rest is used to fixate the head position. To compensate head motion, a head-mounted eye tracker can be used. Here, the camera and the light source are integrated into a helmet and placed in proximity to the eyes, ensuring high-resolution closeup images, which increases the detection rate of features and therefore also the accuracy of the eye tracker. The last variation is a mobile eye tracker, which is a derivative of a head-mounted eye tracker, where the necessary components are still close to the eyes but attached to a lightweight frame like, e.g., a pair of glasses. Furthermore, it is not ensured that the camera and the light source have a constant position, which affects the accuracy. Additionally, the sampling rate of mobile eye-tracking devices is typically between 30 and 120 Hz. This sampling rate is by far lower than the one that can be achieved with modern head-mounted or remote eye-tracking devices, which is currently up to 2000 Hz.

2.2.1 Filtering of Eye-Tracking Data

There are two approaches to process the eye-tracking data. First, the raw recorded gaze data can be used as a basis for visualization and analysis methods, which might get troublesome with an increased sampling rate of the eye tracker, for experiments with many participants, and also for long recordings. Therefore, the second approach aggregates the eye-tracking data before the analysis. The aggregation is achieved by applying a dedicated filter. There are three kinds of filters [34, 35]: *Fixation filters*, *saccade filters*, and *smooth pursuit filters*.

A fixation occurs, if the visuals gaze is maintained in a single location on a stimulus. One of two common methods can be applied to aggregate the raw gaze points to obtain fixations.

Fixation Filter: First, there is the *velocity-threshold identification (I–VT)*. Here, the velocity between successive gaze positions is calculated. As soon as velocity is detected that is below a specified threshold (10–50 °/s), the aggregation of raw gaze points begins. From this point onward, all raw gaze points will be added to the same fixation as long as the velocity stays below the threshold. If the velocity should exceed the threshold, the algorithm stops to aggregate raw gaze points for the current fixation and starts to search for the next beginning of a fixation, respectively the position where the velocity is again below the threshold.

The second approach is based on *dispersion-threshold identification (I-DT)*. Here, instead of a point-to-point velocity, the maximum spatial separation of some successive raw gaze positions is calculated. If the spatial separation is below a threshold, a fixation is generated based on the selected raw gaze points. The selection of the raw gaze points is based on a sliding window approach that defines the minimum number of successive raw gaze points. The window will be extended by one gaze point as long as the dispersion criterion is fulfilled. Additionally, there is a duration threshold (50–250 ms). All fixations that have a duration of less than 100 ms will be discarded.

Saccade Filter: A saccade is defined as a fast eye movement between to successive fixations. A common approach to identify saccades or raw gaze points on saccades is to simply apply a fixation filter and assign all rejected points between two fixations to a saccade. A velocity-threshold-based approach can also be used with a lower threshold of 30–100 °/s. Furthermore, acceleration can indicate the beginning of a saccade, if the acceleration is above a threshold of 4000–8000 $^\circ/s^2$. Using the acceleration is noisy and therefore rather impractical.

Smooth Pursuit Filter: The third category of filters is supposed to detect smooth pursuit, which is a kind of eye movement that is automatically triggered while following moving objects. The detection of smooth pursuit is not easy using threshold-based methods, such as described previously, since smooth pursuit exhibits both fixation and saccadic characteristics: A moving object is fixated, and catch-up saccades occur to follow the object if the gaze position lags behind the object and back-up saccades are used to compensate gaze positions that are ahead of the object. The velocity threshold interval (typically less than 30–40 °/s) for detection of smooth pursuit overlaps with the intervals for the detection of fixations and saccades. Therefore, a threshold-based approach can be extended by an analysis of the direction of gaze position. Using a sliding window approach with window size N, aggregation lines are calculated. Next, the angular difference between successive lines is calculated. Finally, the standard deviation is used to separate fixations from smooth pursuit. A high standard deviation is most likely the result of small eye movements (eye jitter or microsaccades), while a low deviation indicates a rather constant movement of the eyes, most likely in the same direction.

2.2.2 Eye-Tracking Metrics

Metrics are essential for many applications since they are used to quantify characteristics of data, individual data instances, or relationships between them. Based on such abstract representations, analysts are capable of understand large amounts of data fast and fairly reliably depending on the metric, or a combination of multiple metrics. Combinations of metrics are also often the basis for machine learning algorithms. Here, such combinations are called *feature vector*. In the field of eye tracking, metrics are also of relevance, since much data is recorded during an experiment, and the data needs to be analyzed and interpreted. For this purpose, some metrics can be used to infer information related to cognitive processes, workload, and other aspects of the visual system. A large range of metrics is discussed in the book of Holmqvist et al. [36]. They can be categorized into three categories. First, there are movement measures, i.e., for eye-movements based on length and duration of saccades and fixations:

Average saccade duration: Average time to move from one fixation to another and the average time with no visual intake. Average saccade duration is decreasing for more difficult tasks, as well as with a decreased processing capacity.

Average saccade length: Also called saccade amplitude. A long saccade length can be interpreted as an explorative eye movement, whereas short saccade lengths may occur when the task difficulty increases as short eye movements are used to collect information from a restricted area to support the current cognitive process. Furthermore, this metric is often used to quantify similarity.

Average fixation duration: It is often used as an indicator for the cognitive processing depth. High values typically mean that a participant spent more time thinking about an area, e.g., due to high complexity of the scene or absence of intuitiveness in it. Low values in a local area can be the result of stress.

By considering both average saccade length and fixation duration, one can establish a guideline for indicating or interpreting *visual stress*: short average saccade length and low fixation duration might indicate a high level of stress. The second group of metrics contains numerosity measures obtained by, e.g., counting events or other occurrence frequencies:

Number of fixations: General measure that can be applied to specifically defined areas or the whole stimulus, and it is correlated to the time spent in an associated area. The combination of spent time and number of fixations could be used to find different kinds of behavior, e.g., areas that exhibit the same number of fixations but different spent time indicate a different behavior, possibly influenced by the content of the associated area.

Number of saccades: Proportional to the number of fixations and related to the fixation duration. If the number of saccades is increasing, in a fixed amount of time, the fixation duration is decreasing.

$\mathcal{K}$ **coefficient:** Characterizes dynamics of ambient and focal attention per individual scanpath [37]. Positive values indicate longer fixation durations and shorter saccades, therefore focal attention. Negative values indicate the opposite: longer saccades and shorter fixations ambient attention. In the ambiguous case, where the coefficient is zero, subjects could have made either long saccades followed by long fixations or short saccades followed by short fixations. Therefore, distinguishing ambient and focal attention is not possible in this case.

Fixation rate: Is roughly proportional to the inverse of the average fixation duration. This metric can be used to interpret task difficulty. Furthermore, it is used to predict the density of fixations on objects and, therefore, an indicator for measuring mental workload.

Saccade rate: In general, almost identical to the fixation rate. It is an indicator of mental workload, arousal, and fatigue. The saccade rate is decreasing with an increase of task difficulty, mental workload, or fatigue. Arousal leads to an increase of the saccade rate.

Based on such measures, it is possible to determine stressful or maybe difficult situations that participants encounter, e.g., if the fixation rate suddenly exceeds a defined threshold and focal attention is indicated.

The last group contains statistical (position) measures of spatial distributions of fixation for x- and y-direction, which allow getting an impression about the most prominent areas within stimuli:

Standard deviation: Used to quantify the amount of variation or dispersion of a distribution. Lower values indicate that the values of the distribution are closer to the mean value, whereas larger values of the distribution indicate a higher dispersion.

Skewness: Measures the asymmetry of a distribution, i.e., whether either the left tail of a unimodal distribution (negative skewness) or the right tail (positive skewness) is longer or fatter.

Kurtosis: Like skewness, it describes the shape of a distribution. It is the fourth-order moment of the distribution and tells an analyst reasons for the variance within the data. Higher kurtosis indicates infrequent extreme deviations, whereas lower kurtosis indicates frequent modestly sized deviations.

Such simple statistical measures could be helpful to get an aggregated view of the positions of fixations. The skewness indicates whether fixations are located on the left or right (top or bottom) side of the stimulus. The standard deviation indicates the general spread of the fixations. Finally, the kurtosis tells us whether there is a tendency for multiple clusters of fixations or rather a singular occurrence.

Nevertheless, even in early eye-tracking research, it was shown that the viewing behavior strongly depends on the task [29]. Therefore, there is no clear approach on how to interpret eye-tracking metrics. For each task of a study, the guidelines for the interpretation of the metrics may have to be adapted.

2.3 Classical Statistical Evaluation

Statistical testing is a systematic approach to analyze if tested variables have an influence on associated distributions of measured values. P-values are computed to achieve a comparison of distributions and used to reflect some probability that tested distributions are different or not. Historically, a p-value of $p = 0.05$ is assumed to be sufficient to convince an audience that the tested distributions are statistically significantly different, but this significance level can be adjusted depending to the experiment set up and other impact factors. P-values are often associated with a *Type I error*: incorrect rejection of a true null hypothesis. Another error that is rarely reported is the *Type II error*: failure to reject a false null hypothesis.

The statistical testing based on p-values can be categorized into two different classes. If necessary preconditions are fulfilled, *parametric* tests can be used, otherwise *non-parametric* test should be applied. In both cases, measurements (dependent variables) are tested according to a categorization (independent variables) that corresponds to the distributions to test. The independent variables are often referred to as *levels* of associated *factors*. In the example of a drug trial, one factor could be *medicine*, which consists of many drugs (levels) that are going to be tested.

2.3.1 Parametric Testing

A commonly used approach for a parametric test [38] is the analysis of variance (ANOVA), which analyzes the differences among groups by observing the variance to find an explanation for possible differences. There are two necessary preconditions, which are a normal distribution of the data and the homogeneity of variance. The first one can be tested with, e.g., a Shapiro-Wilk test, while the Bartlett test indicates the latter. The basic assumption of an ANOVA is that the group means do not vary beyond the assumed variance, i.e., the normal distributions overlap but do not necessarily need to be equal. This assumption is rejected if the p-value is smaller than a specified significance level. Therefore, there is evidence that not all distributions are from the same population.

The following example shows how to perform an ANOVA based on the following distributions:

Distribution 1 (d1):	39	43	49	48	51	52
Distribution 2 (d2):	37	40	41	45	46	49

The null-hypothesis states that all means are equal and that they are from the same population. Therefore, there should be no difference between the variance of $d1 \cup d2$ and each distribution. Mean values, a sum of squares (SSQ) of the deviations, and degrees of freedom (DoF) are calculated to show this.

	mean	DoF	SSQ
$d1$:	47	5	126
$d2$:	43	5	98
$d1 \cup d2$:	45	11	272
$d1 + d2$:	-	10	224

By considering $d1$ and $d2$ as individual distributions, here referred as $d1 + d2$, the DoF and the SSQ are reduced, and the question is now if this reduction of the DoF explains the reduction of the SSQ (the change of the variance). The final result of the ANOVA can be summarized as followed:

	DoF	SSQ	MSSQ	F	p
Factor:	1	48	48.0	2.14	0.174
Residuals:	10	224	22.4		
Total:	11	272			

The results of the ANOVA are presented in the form $F(DoF, residual) = F$.

Here, the first line is of most interest, since it contains information concerning the difference of the two tested distributions (DoF=1), the difference of the SSQs, the mean sum of squares (MSSQ), the F-value, and the p-value.

The mean sum of squares (MSSQ) is represented by the quotient of $\frac{SSQ}{DoF}$. The F-value is now the important number to determine if there is a difference between the distributions. In case that there is a difference, the value would be $F > 1$, otherwise $F = 1$ if they are the same. Higher F-values indicate a larger probability that the distributions are indeed different. The probability is represented by the p-value, which is estimated by a z-, t, or f-distribution. Based on one these distributions, the DoF, and the F-value, a corresponding p-value can be calculated.

If there are significant results indicated by the test with a DoF > 2, additional t-test between all pairs are required to specify the distributions that differ. This is called a post-hoc analysis.

2.3.2 Non-Parametric Testing

If the necessary preconditions for an ANOVA are not fulfilled, non-parametric tests [39] are used instead to determine significance, e.g., a Kruskal-Wallis test as an initial test followed by pairwise Wilcoxon test as means of a post-hoc analysis. The Kruskal-Wallis test is equivalent to an ANOVA and, therefore, shows that there is at least one distribution that is different from the rest. The Wilcoxon test is the counterpart of the t-test and is used to identify the exact differences between pairs of distributions. The basic idea behind non-parametric tests is to calculate a value that indicates the overlap of the tested distributions. They do not rely on the computations of mean and variance values, instead, ranks are used.

Values:	37	39	40	41	43	45	46	48	49	49	51	52
Rank:	1	2	3	4	5	6	7	8	9.5	9.5	11	12

Given the distribution in the previous example, the Wilcoxon test first merges the distributions and sorts the values of the results in ascending order. If multiple entries with the same values appear, they all get assigned the same average rank. In this case, 49 appears twice and, thereby, a rank of 9.5 is assigned to both. Now all U_i (here with $i \in d1, d2$) are calculated according to Equation 2.1, which are indicators for the amount of overlap between the distributions. Here, R_i is the sum of all ranks of distribution i and is reduced by the minimum rank of U_i, if the ranges of both distributions would not overlap. Here, n_i is the number of elements in distribution i.

$$U_i = R_i - \frac{n_i(n_i+1)}{2} \tag{2.1}$$

A significance value can now be retrieved by a lookup in a χ^2 distribution by using $\min(U_i)$ and the DoF (here 11). The non-parametric test results are commonly presented in the form $\chi^2(DoF) = F$.

In the case that there are more than two distributions, the Kruskal-Wallis test performs the same steps: merge, order, and rank the distributions, followed by a calculation of a descriptive key value H (see Equation 2.2a).

$$H = (N-1)\frac{\sum_{i=1}^{g} n_i(\bar{r}_i - \bar{r})^2}{\sum_{i=1}^{g}\sum_{j=1}^{n_i}(r_{ij} - \bar{r})^2} \tag{2.2a}$$

$$\bar{r}_i = \frac{\sum_{j=1}^{n_i} r_{ij}}{n_i} \tag{2.2b}$$

$$\bar{r} = \frac{1}{2}(N+1) \tag{2.2c}$$

Here, g is the number of distributions, n_i the number of values in distribution i, r_{ij} the rank of values j in distribution i. The average rank of all values in distribution i is represented by $\bar{r}_i$, and $\bar{r}$ is the average of all r_{ij}.

The non-parametric test indicates here the same result as the parametric test: there are no significant differences ($\chi^2(1) = 1.86; p = 0.17$).

2.4 New Statistics

In contrast to the previously described traditional way to perform a statistical analysis and report the results by drawing conclusions about a population based on sample data, a concept called *new statistics* [40] is increasingly adopted, which relies on performing statistical analysis without heavily relying on null-hypothesis testing and p-values. The idea behind this concept is to report properties of distributions of data that should support an analyst in interpreting the results. It is also said that this way of reporting results matches the natural way how humans think about results since there are no hard representations of the results (binary decision of statistical significance of results depending on the p-value and its threshold). Instead, the concept of *new statistics* provides room for interpretations to answer the question of *how much of an effect there is* in contrast to null-hypothesis testing that directly tries to answer the question *if there is an effect or not*.

Contrary to the name, this is not a novel way of performing a statistical evaluation, in fact, it is already employed in many disciplines, like chemistry or physics, where, e.g., often point estimates of population parameters are reported in combination with interval estimates. Most commonly, these are the mean value of a sample distribution and an associated confidence interval, which is calculated according to Equation 2.3.

$$CI = \left[E(X) - z_{1-p}\frac{\sigma}{\sqrt{N}}; E(X) + z_{1-p}\frac{\sigma}{\sqrt{N}}\right] \tag{2.3}$$

Here, $E(X)$ is the expected value of the random variable X (distribution), commonly the mean value, which is used to calculate the upper and lower bound of the confidence interval (CI). The value that is subtracted and added to $E(X)$ is based in the standard error, which is the standard deviation σ divided by square root of the size of the distribution $\sqrt{N}$. The actual size of the confidence interval is specified by z_{1-p} and is related to the p-value. A p-value of 0.05 corresponds to a confidence level of 95% and this is used to calculate the number of standardized deviations that are needed to cover 95% of a z-distribution, which results in this case in a value of 1.96. Using a z-distribution assumes that the data are normal distributed. If this is not the case, a t-distribution can be used instead to calculate the corresponding scaling value $t_{DoF,1-p}$. For both cases, there are lookup tables to retrieve z or t values for typical confidence levels.

Another means to report statistical data is the effect size, which is a quantitative measure to express the strength of the effects of a tested parameter. There are different ways to express the effect size, e.g., correlation, stating risks, or comparing the effects of a specific treatment to the global measured data of an experiment. One important factor is here to make results comparable, which is achieved by scaling the effect-size value with respect to the standard error, which may be computed differently depending on the used type of effect size.

Therefore, the generation of confidence intervals also depends on the used type of effect size. Examples of effect size types are, e.g., Pearson's correlation coefficient or the Eta-squared measure.

New statistics do not only rely on numbers as previous explanations might suggest. Diverse plots can be used to give an analyst an overview of the distribution and additionally encode confidence intervals. Here, simple bar charts can be used to show mean values, and confidence intervals or box-plots can depict the distribution of the data based on quantiles. Q-Q plots are also often utilized to identify a possibly significant difference between two tested distributions visually.

2.5 Visualization and Analysis

Eye tracking is nowadays used in a variety of disciplines reaching from marketing to medicine and also visual research. Regardless of the domain eye tracking is used in, analysts have to deal with a huge amount of data. Consider a small study that had 20 participants each confronted to solve a task for 100 stimuli. That would result in 2,000 scanpaths that have to be analyzed. Let us further assume that each stimulus is presented for 30 seconds, therefore, with current high-end eye-tracking devices with a binocular sampling frequency of 2000 Hz, this would result in 60,000 recorded gaze points for one stimulus and overall in 120 million recorded gaze points. Based on a conservative estimate, this number can be reduced to approximately 120,000 fixations after applying a fixation filter. This highlights a first problem in eye tracking, which is a large amount of data that needs to be handled and analyzed. A second problem is that there is a wide field of applications for eye tracking, leading to different kinds of research questions. Hence, often different or specialized analysis approaches and algorithms are required that not only focus on completion time and task accuracy.

The summary of the previous example and statements is that analysts need support in analyzing the data, which is where visualization comes into play. Here, descriptive images are generated that show properties of the data. If visualization is combined with interaction methods that allow an explorative analysis of the data to find structures by relying on perceptual and cognitive abilities of humans, methods are created that are categorized as *visual analytics*.

Up to now, a variety of such methods has been developed. Blascheck et al. [41] established a taxonomy and categorized a variety of eye-tracking visualization and evaluation methods. There are three overarching categories, which are point-based, AOI-based, or a combination of both.

Point-based methods usually focus on the eye movements and their spatial or temporal distribution and consist of, e.g., timeline visualizations, attention maps, scanpath visualizations, or space-time cubes. These methods are not always sufficient, e.g., in case of asynchronous data or if the analysis requires to consider a semantic context. For both cases, AOI-based methods are a way to address these issues, since each AOI has a semantic, e.g., a description of the covered area. Furthermore, transitions and relations between AOIs are the focus of AOI-based evaluation methods. With a sequence analysis, the problem of asynchronous data becomes less relevant. AOIs are often used in timeline AOI visualizations, or relational AOI visualizations, which include transitions, sequences, and hierarchies of AOIs. Naturally, point-based and AOI-based approaches are not mutually exclusive and can be utilized simultaneously. Additionally, the visualizations can be divided into several sub-categories: temporal, spatial, spatiotemporal, animated, static, single user, multiple users, 2D, or 3D.

Part I

Methodology for Data Analysis

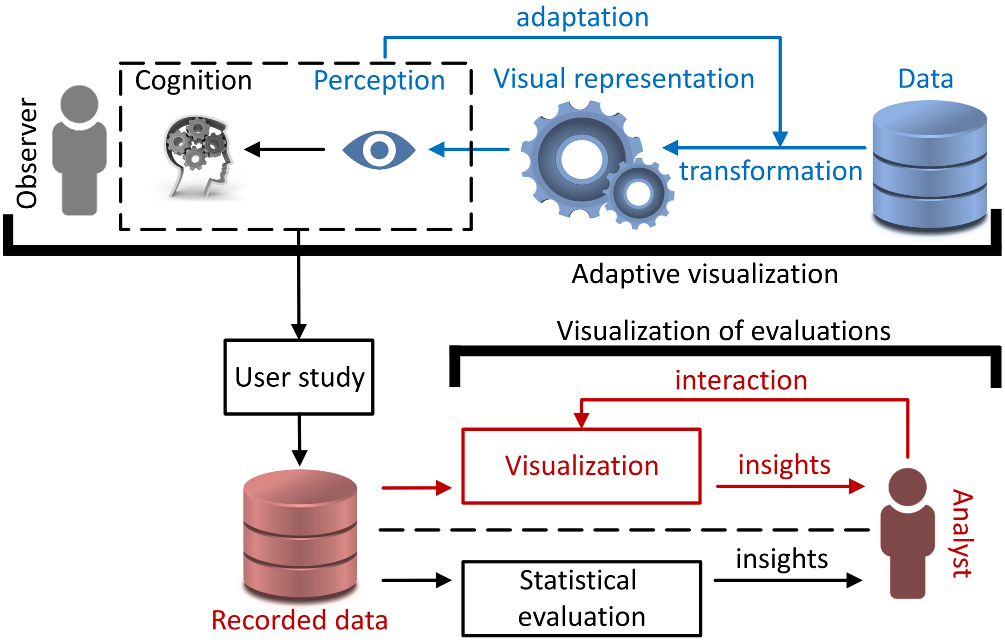


Figure I — Components and workflows involved in Part I.

Regardless of the domain, data is being generated, and it needs to be analyzed. Often this is possible without great effort, e.g., if there are only a few numbers that can be checked manually or by using, e.g., a simple diagram to summarize the data more appealingly. Looking at a larger amount of data, it is not always possible to understand the data or to draw conclusions from them. Here, more complex and also structured processes are needed to perform data analysis. Part I of this thesis addresses two different approaches for the analysis of data.

The first is an automated analysis of data that is transformed to represent the data in a more comprehensible manner, e.g., using an image. Such image-based representations are now analyzed based on perceptual models and depending on their output, an adaptation of the representations might occur to improve the perception of them by, e.g., compensate illusionary effects.

With respect to the blue highlighted parts in Figure I, Chapter 3.2 presents a method to perform a quality assessment of advection visualization techniques based on a perceptual model. In Chapter 3.3 perceptual models are now used to quantify the tilt illusion effect and perform an appropriate compensation.

The second approach is based on classical visualization depending on an analyst to utilize an analysis framework in order to gain insights into the data. In this particular case, eye-tracking data are used that were recorded during user studies. More precisely, an eye tracker was used to record eye-gaze positions on presented stimuli. This is reflected in the red highlighted parts in Figure I and corresponding methods are proposed to address problems related to eye-tracking: data annotation of eye-gaze data (Chapter 4.2), a static spatio-temporal visualization of gaze data (Chapter 4.3), and the utilization of multiple eye-tracking metrics to group participants that exhibit similar gaze behavior (Chapter 4.4).

Chapter 3

Model-based Visual Quality Assessment

In many scenarios, models are used to analyze or interact with data ranging from models obtained through machine learning to models that exploit the results of experiments that measured perceptual effects. This chapter[1] addresses applications that utilize models for quantification, beginning with diffusion effects of advection methods. Here, a key value is being calculated based on a spectral analysis that is used to show the loss of high frequencies with the progress of an advection. Next, models are used first to extract perceived directions of textures and afterward to compensate for illusions of incorrectly perceived directions. The simultaneous orientation contrast causes this phenomenon: frequencies of different textures influence each other, leading to an incorrect perception. This is related to simultaneous color contrast, where colors are perceived differently, depending on their surroundings.

3.1 Related Work

Research data can be analyzed in different ways, e.g., by utilizing or establishing models. They can either be used to in automatic analysis processes to handle

---

[1] **Parts of this chapter have been published in:**
R. Netzel, M. Ament, M. Burch, and D. Weiskopf. Spectral analysis of higher-order and BFECC texture advection. In *Workshop on Vision, Modeling and Visualization (VMV)*, pages 87–94, 2012 [1], © 2012 Eurographics.

large amounts of data or for the generation of representations of data for visual analysis. One possible scenario is to visualize the trend of a vector field. Sophisticated methods integrate an underlying vector field and, thereby, transport a quantity that is then visualized. The quality of such methods can be quantified based on established models (see Section 3.1.1). Another example of utilizing models is to compensate for effects, i.e., simultaneous contrast or other illusions that are introduced due to characteristics of the human visual system (see Section 3.1.2).

3.1.1 Visualizing Vector Fields

A simple method to get an impression of the trend of a flow is to put glyphs, like arrows, into a given image. They will show then the direction of the flow at respective positions. The problem here is the size of the glyphs. The bigger they are the better they can be recognized but the level of detail of the flow that can be illustrated decreases. Another idea is to generate a texture that describes the flow with the needed level of detail. For this purpose, Line Integral Convolution (LIC) [42] was introduced. This method takes a white-noise image and filters it along the streamlines of an underlying vector field. Another approach for texture generation is the Spot Noise method [43]. Colored spots are randomly placed and smeared according to a vector field. In contrast to methods where intensity values are advected, flows can also be visualized by a source that emits dye that is then transported by the flow [44, 45]. Furthermore, advection methods can be used to imitate high-resolution fluids [46].

All methods of texture advection can be grouped into two views. The first one is the Lagrangian view in which a computation area is regarded as a particle system with particles at every location. The particles in this system can be moved according to a vector field [47]. In the semi-Lagrangian view, which can be considered as an iterative method, the particles are just moved one time step to get intermediate data [48, 49, 50]. Techniques can be categorized into forward or backward strategies. Forward means that the field is integrated forward in the direction of the field vectors to calculate the new positions of particles [49, 45, 51]. Backward integration methods perform a step back in time to compute the origin for every position in a given image or for every particle in a system. Values are computed by means of interpolation at the original positions and then assigned to the new one [52, 53, 49, 45]. A combination of both strategies is possible [54, 55] to increase the accuracy of the advection. Furthermore, high-quality advection can be achieved by means of physically-based advection, e.g., with area sampling techniques [44] or finite volume approaches [56]. However, these methods are computationally expensive.

For the integration of a vector field, there is a variety of procedures. A simple one is the Euler method, which computes the next location according to a first-order approximation. This method can integrate the vector field correctly if the time step converges to zero. Otherwise it might diverge. A second-order approximation is implemented by the Leap-Frog method [57], which works more accurately than the Euler method but might divagate if the time step is too large. The Runge-Kutta method shows even better results than the Leap-Frog approach and can be implemented as an approximation of fourth or even higher order [58].

3.1.2 Perceptual Effects and Compensation

When it comes to creating a visualization, there are many factors that need to be considered, e.g., which graphical primitives or color encoding should be used. Often such design aspects are chosen at first without considering the impact on perception and possible problems or phenomena that could occur due to the physiology of the human visual system.

One of these problems is the simultaneous color contrast, which is a visual illusion where one color is perceived differently depending on its surrounding colors. By performing experiments, it is possible to derive models that approximate such phenomena mathematically or algorithmically. To this point, Mittelstädt et al. [59] developed an algorithm for the compensation of the simultaneous color contrast by optimization of perceived color. Here, the colors of a background and a foreground texture are adjusted in an iterative process. In each iteration, the background texture is evaluated based on a color appearance model (iCAM) and adjusted accordingly. In subsequent work, Mittelstädt et al. [60] extended the approach to account for interpersonal differences. Another effect related to overlaid textures is the tiled illusion. Depending on the difference of the perceived main direction of a background and a foreground texture, the texture on top could appear tiled. Schwartz et al. [61] detected the patterns of tilt illusions in natural images by spatially limited search fields and measured the perceived tilt. In other research, it was shown that the spatial frequency, as well as the contrast, influence the perceived tilt bias. To this end, Georgeson [62] reported the tilt bias for different spatial frequencies of foreground and background texture. Durant et al. [63] studied the tiled effect by presenting foreground and background asynchronously, while also varying the contrast of both. Some illusions, like the tilt illusion or illusionary contours, are caused in the first two layers (V1 and V2) in the visual cortex (see Section 2.1). It was shown that the receptive fields of so-called *simple cells* are involved, which detect edges by overlapping receptive fields and lateral inhibition.

Wei et al. [64] modeled such simple cells as Gabor patches and used them to identify line segments that seem to deform when lines are crossing, but there no compensation was performed. Franceschiello et al. [65] also used this model for simple cells to derive a model that specifically compensates Hering illusions.

3.2 Frequency Spectrum and Visual Quality

Advection methods are widely used for visualization in different domains, e.g., to show the flow in fluids, air, or electromagnetic fields. Such phenomena are usually described by vector fields that can be visualized by computing descriptive representations of transport mechanisms.

Regardless of the used method, the aim is always to solve the advection equation so that important features, such as edges or the spectrum of the resulting image, are conserved. However, discretization and interpolation lead to errors that occur while solving the equations. This leads to numerical diffusion and consequently to a loss of the feature properties. Especially iterative methods implicitly implement a low-pass filter while providing a very fast computation. Therefore, the analysis of the spectral behavior is of relevance. To this end, Yu et al. [51] demonstrated that their Lagrangian method preserves the spectrum of the input texture. Furthermore, Weiskopf [66] had a look at the spectral appearance of an exponential filter kernel in the context of LIC flow visualization. However, so far no one has shown the exact effect of grid-based advection methods in frequency space by regarding a spectral change.

The radial power spectrum diagram (RPSD) is used to address this issue, for quantification of diffusion effects and a comparison of different advection methods is proposed in this section. The methods used for comparison and analysis are backward and BFECC integration strategies as well as their combination with higher-order interpolation methods. Furthermore, the influence of the space discretization is investigated by testing different resolutions of a texture that is advected.

The method proposed addresses the sub-workflow highlighted in blue in Figure I, but with the restriction that only a key value is computed without using it to perform any adaptation.

3.2.1 Linear Advection

Linear advection is the process of passive transport of a scalar quantity $\varphi(x,t)$ within a vector field $v(x,t)$. In the Eulerian frame, the process of advection is described by a continuity equation:

$$\frac{\partial \varphi}{\partial t} + \nabla \cdot (\varphi \boldsymbol{v}) = 0. \tag{3.1}$$

The scalar quantity φ describes the distribution of the advected material, e.g., the density of smoke or, as in this case, the color intensities $\boldsymbol{\varphi} = (c_R, c_G, c_B)$ of an RGB texture. As the channels are independent of each other, three independent advection equations are obtained:

$$\frac{\partial \boldsymbol{\varphi}_i}{\partial t} + \nabla \cdot (\boldsymbol{\varphi}_i \boldsymbol{v}) = 0, \qquad i = 1,2,3, \tag{3.2}$$

one for each channel. However, the numerical solution of Equation (3.2) typically introduces an error term $\boldsymbol{e}_i(\boldsymbol{x}, t) \neq 0$ on the right-hand side:

$$\frac{\partial \boldsymbol{\varphi}_i}{\partial t} + \nabla \cdot (\boldsymbol{\varphi}_i \boldsymbol{v}) = \boldsymbol{e}_i, \qquad i = 1,2,3. \tag{3.3}$$

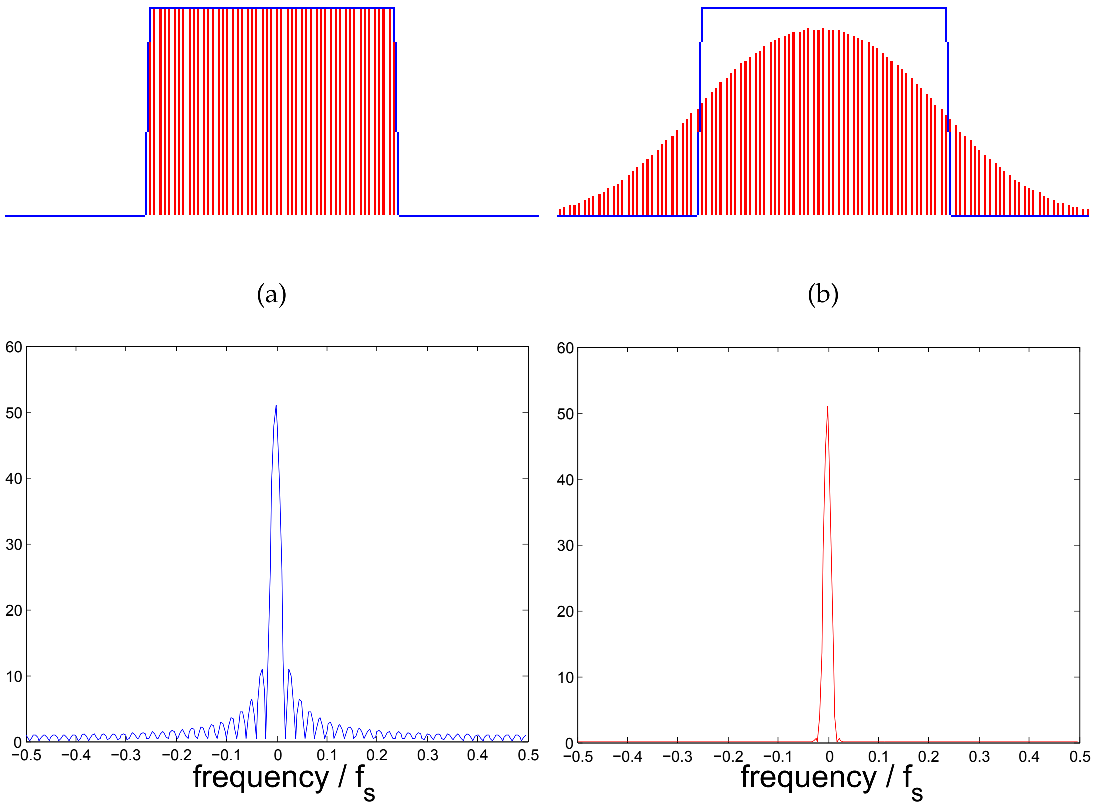
Depending on the solution method, the error term is dominated by different side effects, such as numerical diffusion or dispersion [67]. In the following, the focus is on numerical diffusion, which is usually dominant in Eulerian and semi-Lagrangian-based advection algorithms as the consequence of repeated, usually linear, interpolation operations. Figure 3.1 depicts the impact of numerical diffusion in a 1D example of an advected box function, using the semi-Lagrange method. In this case, a modified partial differential equation is solved, i.e., an advection–diffusion equation:

$$\frac{\partial \boldsymbol{\varphi}_i}{\partial t} + \nabla \cdot (\boldsymbol{\varphi}_i \boldsymbol{v}) = \underbrace{D \Delta \boldsymbol{\varphi}_i + \boldsymbol{f}_i}_{= \boldsymbol{e}_i} \qquad i = 1,2,3, \tag{3.4}$$

where D is a diffusion coefficient and $\boldsymbol{f}_i$ denotes other sources of errors. However, it is assumed that $|\boldsymbol{f}_i| \ll |D \Delta \boldsymbol{\varphi}_i|$, which is a known fact for semi-Lagrangian-based methods [55]. The following sections make use of this assumption. In the special case where the vector field $\boldsymbol{v}$ is solenoidal, e.g., if $\boldsymbol{v}$ is the solution of the incompressible Navier-Stokes equations, Equation (3.4) becomes simpler:

$$\frac{\partial \boldsymbol{\varphi}_i}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{\varphi}_i = D \Delta \boldsymbol{\varphi}_i + \boldsymbol{f}_i, \qquad i = 1,2,3. \tag{3.5}$$

However, independent of the properties of the vector field, the additional diffusion term implements a low-pass filter and the magnitude of the diffusion coefficient determines the amount of smoothing. Depending on the algorithm, the diffusion coefficient may be a function of different quantities, and its



(a) (b) (c) (d)

Figure 3.1 — Numerical diffusion in one dimension. (a) Analytical box function (blue) and discrete samples (red). (b) Semi-Lagrange advection after 50 time steps. In the spatial domain, the discrete samples are smoothened across the sharp boundary of the analytic solution. (c) Fourier transformation of the box function. (d) The Fourier transformation of the numerical solution after 50 time steps shows significant damping of high frequencies. Frequencies are normalized with respect to the sample frequency f_s.

magnitude can vary significantly. While it is possible to find analytic expressions for the diffusion coefficient for simple techniques, such as the semi-Lagrange method in one dimension [68], it is difficult to derive an analytic expression in the general case, especially when limiters and higher-order interpolation schemes are combined in two or more dimensions as in this case. Therefore, a spectral analysis is employed to quantify the amount of numerical diffusion by studying the conservation of high frequencies over the advection time.

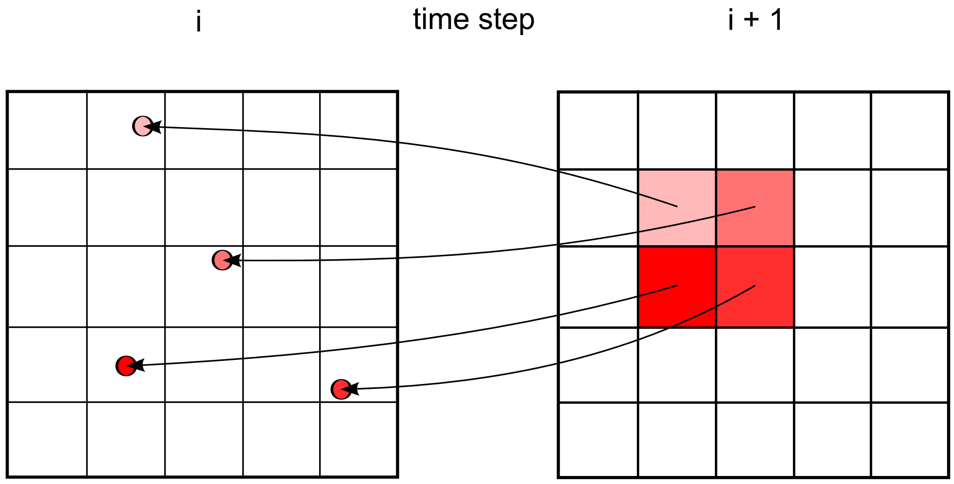


Figure 3.2 — Backward integration with interpolation of intensity values.

3.2.2 Advection Techniques

In this section, common basic techniques are described that are used to compute advection, and the basic concept of higher-order interpolation is briefly discussed.

Ground Truth

For the spectral analysis and comparison, a ground truth is required that does not exhibit numerical diffusion. Thus, to compute the color of each pixel after a finite advection time $t = T$, a full path line is integrated backward in time to determine the location of each pixel in the original image at advection time $t = 0$ [52].

$$\boldsymbol{x}(0) = \boldsymbol{x}(T) + \int_{T}^{0} \boldsymbol{v}(\boldsymbol{x}(t), t) dt. \tag{3.6}$$

Backward

Semi-Lagrangian-based techniques rely on iterative backward integration and interpolation. A backward integration can be specified by to compute the intensity that would be advected to a certain location:

$$\boldsymbol{x}(t_i) = \boldsymbol{x}(t_{i+1}) + \int_{t_{i+1}}^{t_i} \boldsymbol{v}(\boldsymbol{x}(t), t) dt. \tag{3.7}$$

In contrast to forward integration, backward integration does not produce holes in the image of the current time step. There is also no need for scattering intensity values around adjacent pixels, but an interpolation of the transported intensity is required, which is shown in Figure 3.2.

BFECC

The Back and Forth Error Compensation and Correction (BFECC) method was described by Dupont and Lui [69, 70]. This method is a combination of a forward and backward technique and consists of four steps. Backward integration generates the first image. With this image, a forward integration is performed with a second image as a result. A third image is computed, which is the difference of the first and second one. It represents the error that has occurred during the back- and forward integration. In the last step, the intensity values of the original image are modified by adding half of the calculated error to the corresponding intensity of the original image. Based on the corrected image, a backward step is performed to get the final result image.

Higher-Order Interpolation

For the quality of the result of an advection method, it is of great importance to handle transportation of intensities with sub-pixel accuracy. Although bilinear interpolation is attractive due to its high performance, it exhibits high numerical diffusion and causes strong blurring of details. A higher-order interpolation with polynomials of degree three and five is employed to increase the precision of the interpolation. This is depicted in Figure 3.3. For a polynomial of n-th degree, $(n+1)^2$ points are required in 2D around the position $\boldsymbol{x} = (x, y)^T$ that should be interpolated. Then, for each row in the resulting square, a polynomial is interpolated and evaluated at a position that is equal to the x-component of $\boldsymbol{x}$. With the $n+1$ intermediate data, another polynomial is interpolated and evaluated at a position that is equal to the y component of $\boldsymbol{x}$. This leads to the final intensity value of position $\boldsymbol{x}$.

BFECC with Higher-Order Interpolation

BFECC and higher-order interpolation are orthogonal concepts that have been employed separately in previous approaches to reduce numerical diffusion. However, a combination of both techniques has not been discussed before in the literature. Therefore, all bilinear interpolations are substituted that are commonly used in BFECC methods with higher-order interpolation schemes of degree three and five. In the following section, the impact on quality of both concepts is analyzed and quantified separately and in combination to provide a thorough reference.

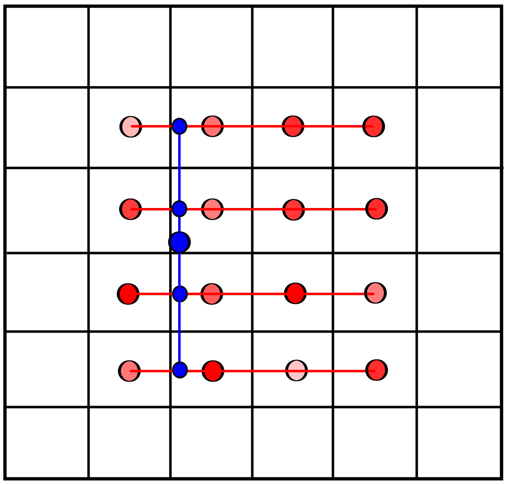


Figure 3.3 — Higher-order interpolation with polynomial of third degree. Colored dots are equals to scalar values. The big blue dot corresponds to the point for that a value is interpolated. Red and blue represent the two phases of interpolation. First, the red values are used for $n + 1$ polynomials that evaluated to get intermediate values (smaller blue dots). In the next step, these values are used to compute the final value.

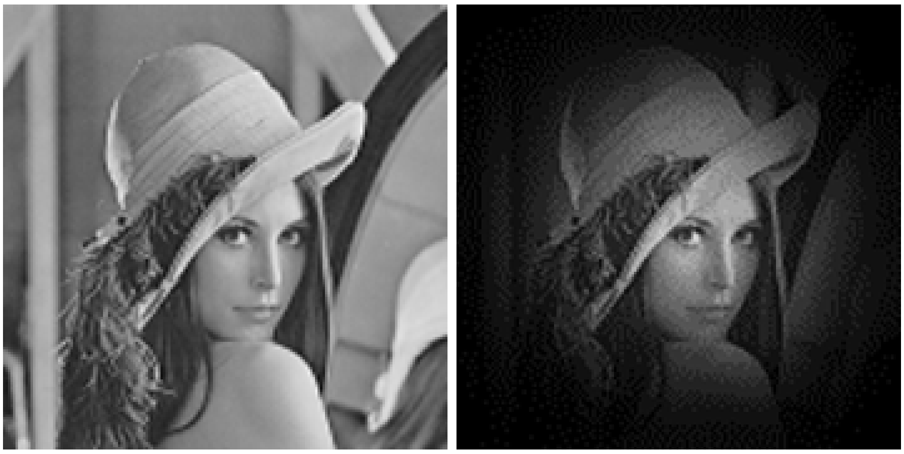
3.2.3 Spectral Analysis Process

The process for analyzing the spectrum of an image consists of four steps. First, a Hamming window on the input image is applied to avoid the artificial generation of spurious high frequencies that could appear due to the periodicity of the image in frequency space. This is shown in Figure 3.4.

Next, a radial power spectrum diagram (RPSD) is computed. Therefore, for each radius, the average power is calculated along a concentric circle located around the zero frequency origin of the frequency transformed image. This is depicted in Figure 3.5. At last, the radius of a circle that covers 99% of the frequency is calculated, and this border is highlighted in the diagram. This can be regarded as a key value of the spectral stability.

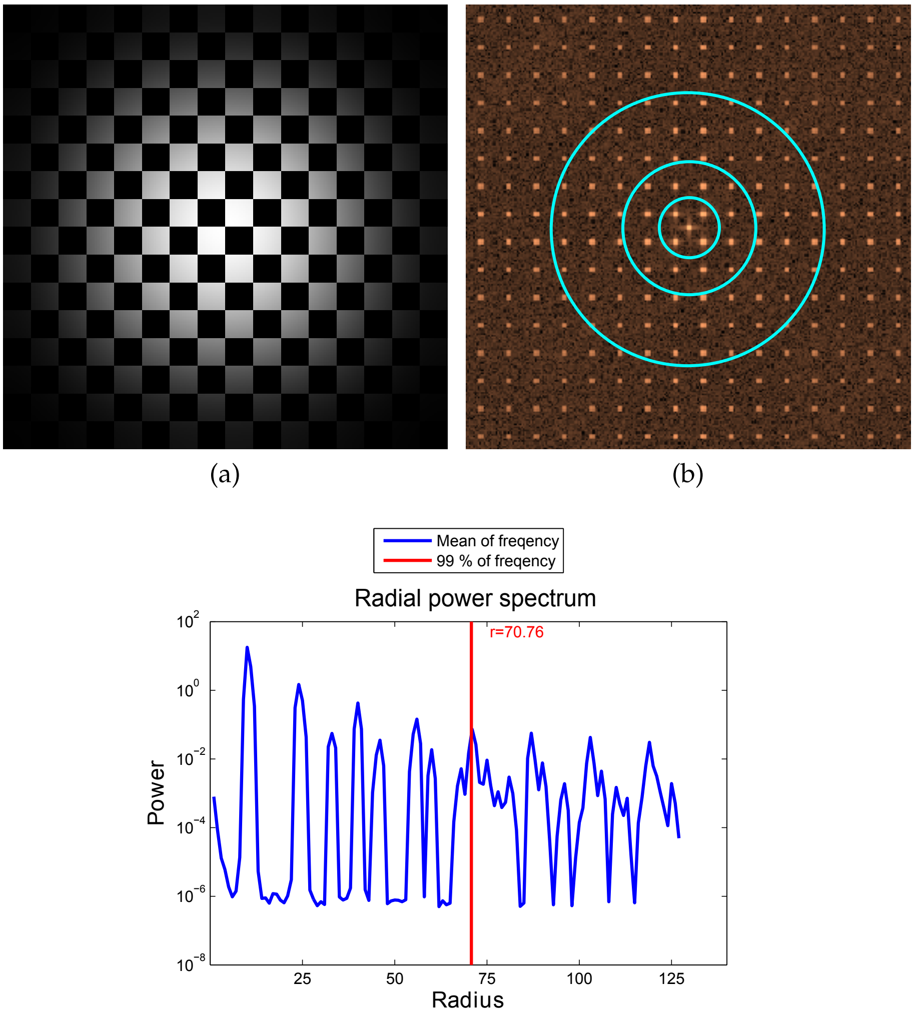
3.2.4 Preliminary Notes

For the evaluation, two input images are used: a checkerboard and a Gaussian filtered white noise image to cover a wide range of different spectra. The checkerboard image is a very regular pattern with distinct high peaks in the spectrum, whereas the noise image is smoothly varying in the Fourier domain and is typically employed for LIC. Both textures were advected with a circular vector field and with a Navier-Stokes vector field. For the integration of the



(a) (b)

Figure 3.4 — (a) shows the input image, and (b) after applying the Hamming window.



(a) (b)

(c)

Figure 3.5 — (a) shows the input image, (b) is the frequency transformed image with some concentric circles while the mean of the integration along them is shown in (c).

vector fields, the length of one time step $\delta t = 0.05$ is chosen to avoid the generation of artifacts through an integration of the vector field. The advection is simulated for 500 time steps and is evaluated after 25, 50, 100, 250, and 500 steps. Additionally, the influence of the resolution of input images on the advection is investigated. For this purpose, a Gaussian filtered white noise image of size $64x64$ is generated. Furthermore, this image is modified in frequency space by cutting off all frequencies that are located at a distance of more than 32.

Zero padding is performed three times in the Fourier domain for upsampling to generate textures of size $128x128$, $256x256$, and $512x512$.

3.2.5 Results

In this section, the results of the evaluation of backward and BFECC methods in combination with different interpolation strategies for texture advection are presented, to demonstrate the magnitude of spectral change caused by blurring. This effect is shown by a decreasing calculated radius in the RPSD for a progressing advection. A small radius indicates that low frequencies are dominant in a texture, which results in blurring. Larger radii indicate containment of higher frequencies, which leads to a better spectral preservation. Hence, image features are conserved better and the blurring is less distinct. This is shown in Figure 3.6 for a circular vector field and a checkerboard texture with resolution $256x256$. A Gaussian filtered noise image with the same resolution was used in Figure 3.7. By comparing the radii of the RPSDs, calculated after an advection over 50 and 500 time steps, it can be seen that by increasing the interpolation order the radii are also increasing and the blurring effect is reduced.

Table 3.1 and Table 3.2 contain the radii for all time steps. They show a decreasing of the radii with the progressing advection for all methods and also the increasing radius values of the methods in each of the time steps. Table 3.3 contains the radii for the checkerboard and the noise image after 100 time steps of advection with a Navier-Stokes vector field (see Figure 3.8). Here one can see that the higher-order strategies perform better.

An evaluation based on the Navier-Stokes vector field with more than 250 timesteps was not possible due to artifacts that were produced by the BFECC methods. This is depicted in Figure 3.8. The artifacts cause high frequencies that lead to an artificial increase of the radii.

Next, the convergence speed of methods is discussed that use interpolation of different orders. The input images for the advection are generated by upsampling a Gaussian filtered noise image as described in the preliminary

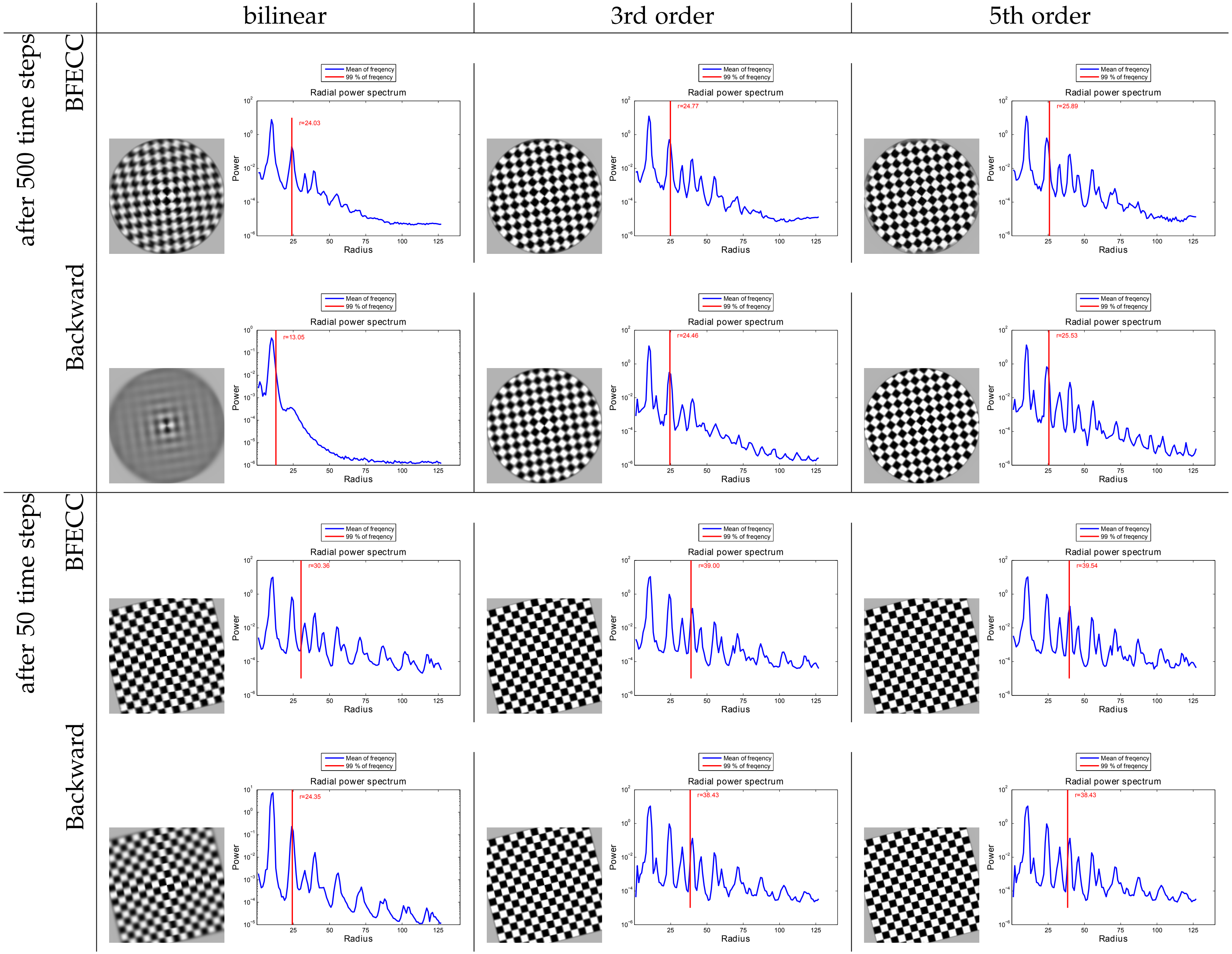


Figure 3.6 — Advection images and radial power spectrum diagrams after 50 and 500 time steps for backward and BFECC-based methods with different interpolation strategies. A checkerboard was used as an input image. The vector field is circular.

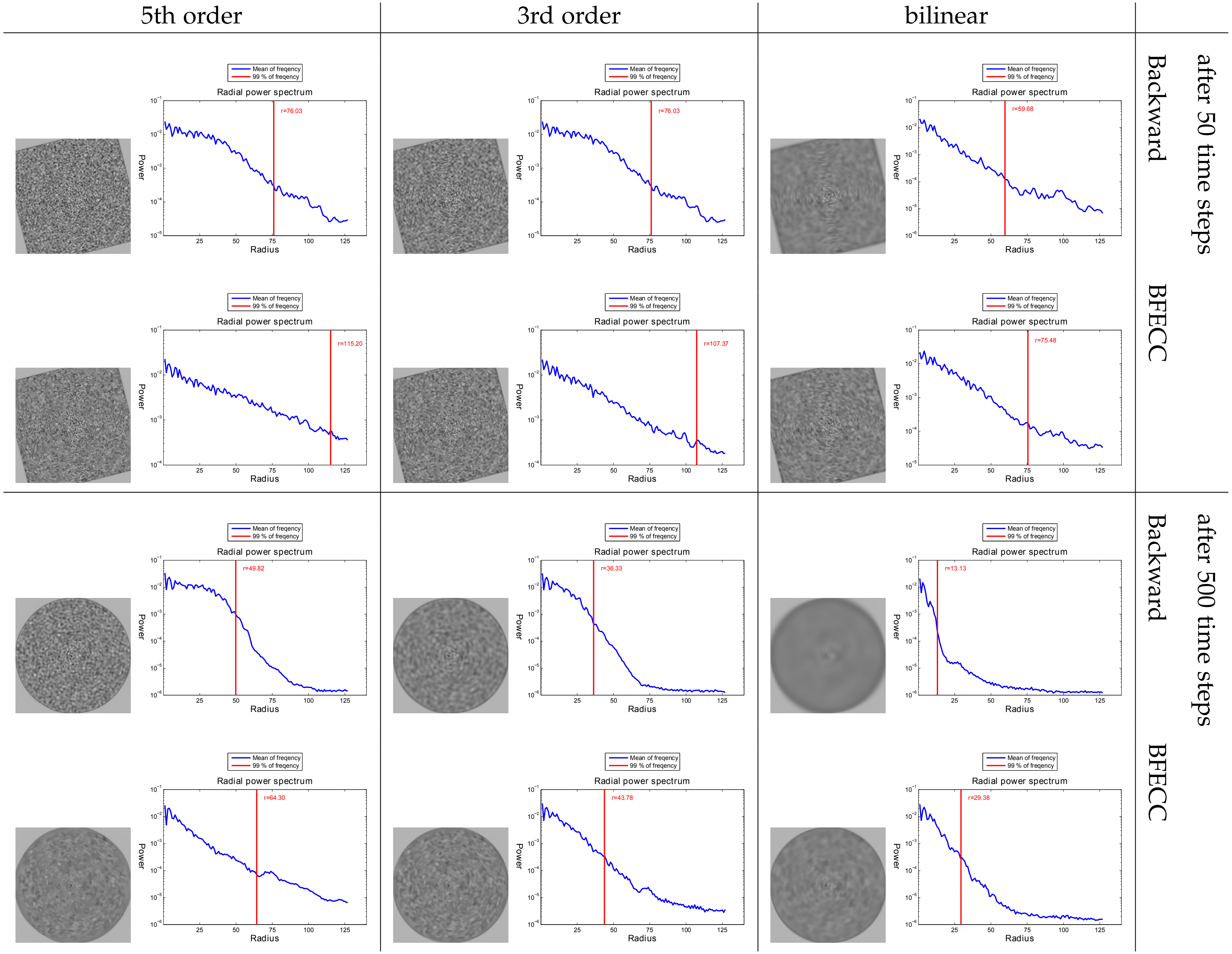


Figure 3.7 — Advection images and radial power spectrum diagrams after 50 and 500 time steps for backward and BFECC-based methods with different interpolation strategies. A Gaussian filtered white noise image is used as an input image. The vector field is circular.

Table 3.1 — Radius for containment of 99% of frequency for all tested methods and time steps. Checkerboard image and a circular vector field are used.

	time steps	25	50	100	250	500
	Ground Truth	55.1	54.7	54.9	55.3	55.4
Backward	bilinear	24.7	24.4	23.5	11.9	13.1
	3rd order	39.3	38.4	24.9	24.7	24.5
	5th order	39.6	39.5	39.1	38.1	25.5
BFECC	bilinear	39.0	30.4	24.5	24.4	24.0
	3rd order	39.4	39.0	32.5	24.9	24.7
	5th order	39.7	39.5	39.2	38.3	25.9

Table 3.2 — Radius for containment of 99% of frequency for all tested methods and time steps. Gaussian filtered noise image and a circular vector field are used.

	time steps	25	50	100	250	500
	Ground Truth	118.6	118.9	118.1	118.7	118.6
Backward	bilinear	83.7	59.7	42.5	25.1	13.1
	3rd order	92.6	76.0	58.9	43.9	36.3
	5th order	98.1	85.8	70.8	56.3	49.8
BFECC	bilinear	93.5	75.5	56.0	38.9	29.4
	3rd order	114.6	107.4	92.3	57.6	43.8
	5th order	118.5	115.2	107.9	87.7	64.3

Table 3.3 — Radius for containment of 99% of frequency for all tested methods after 100 time steps for a Guassian filtered noise image and a checkboard image. A Navier-Stokes vector field is used.

		100 time steps	
		Checkboard	Noise
	Ground Truth	118.1	122.7
Backward	bilinear	42.5	72.7
	3rd order	58.9	78.9
	5th order	70.8	92.5
BFECC	bilinear	56.0	72.6
	3rd order	92.3	113.1
	5th order	107.9	123.8

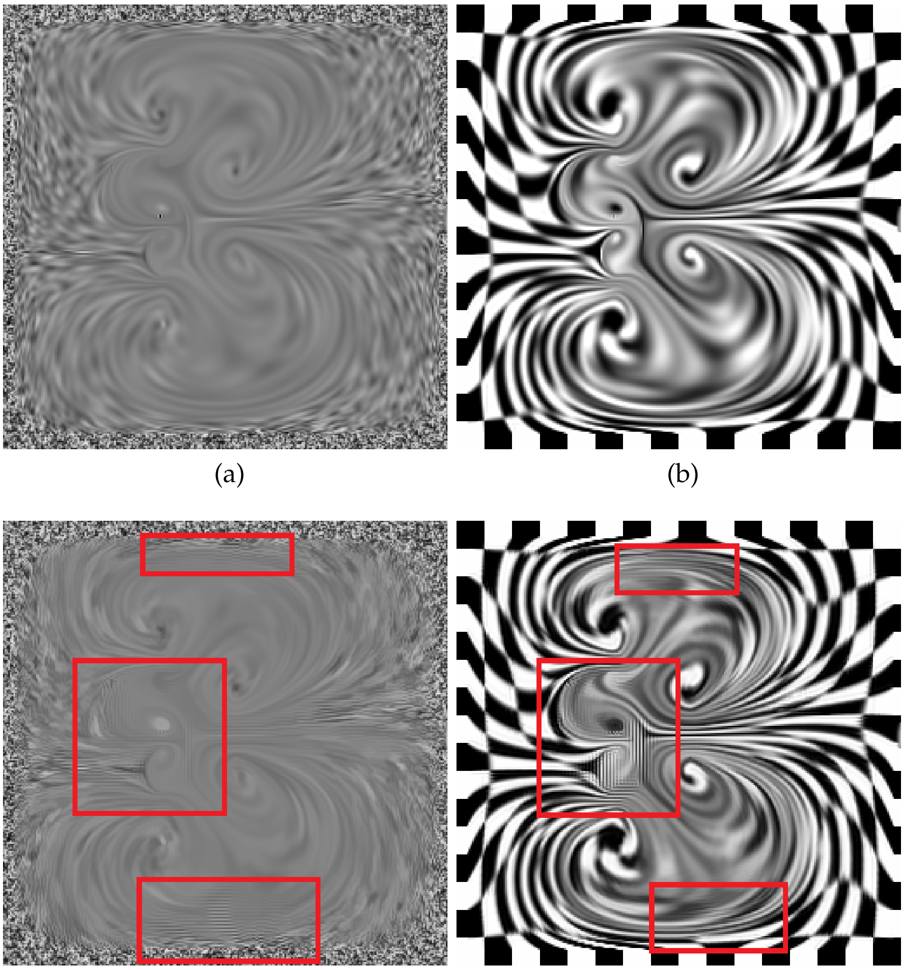


Figure 3.8 — (a)–(d) illustrate an advection process after 500 time steps with interpolation of 3rd order. (a) and (b) are based on a backward method and have no artifacts. (c) and (d) are based on BFECC with artifacts that are highlighted with red rectangles.

notes (Section 3.2.4). The images are advected 500 time steps. Figure 3.9 shows convergence curves for all combinations of methods. These curves represent the ratio of the ground truth radius and the calculated radius for each combination. One can see that higher-order interpolation leads to a faster convergence when increasing the resolution. Additionally, the use of a BFECC strategy influences the convergence positively compared to the backward strategies. However, the BFECC strategies cause artifacts, but the influence of the artifacts decreases with higher resolutions. This can be seen in the ratio values of the BFECC curve with 5th order interpolation, which are at the beginning greater than one.

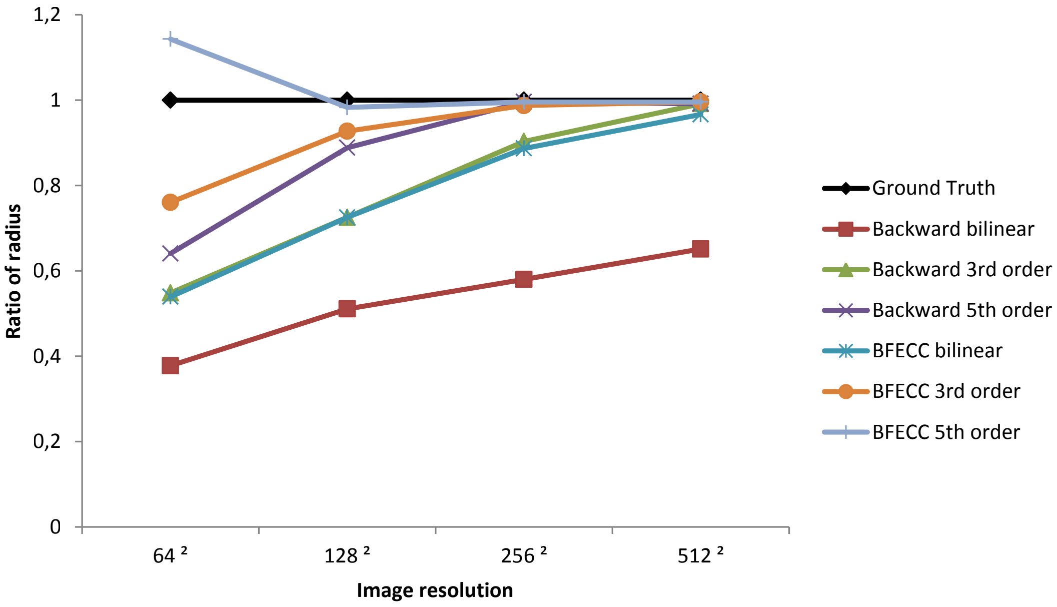


Figure 3.9 — Convergence speed of methods toward the ground truth over different image resolutions. Values are ratios between the actual radius and the ground truth radius after 500 time steps of advection.

3.3 Simultaneous Orientation Contrast

Visualization is used to generate representations of data that are more understandable than the original data in its original form. Therefore, a multitude of different methods was developed for a variety of domains. For instance, in flow visualization to make the structure of vector fields visible, as discussed in the previous section, or in network visualization to show relationships between elements represented as node-link diagrams. Such visualizations often suffer from visual clutter, which might influence the perception and thereby also the interpretation of the visualized data. This is especially true if relevant information is added in a post-processing step to an existing visual representation by superimposing the information on the original representation. Wide-spread examples of such overlays are points of interest and routing instructions on top of topological maps, that help users to keep contextual awareness.

In general, this is related to the topic of superimposing textures. In some cases, overlaying textures may produce a perceptually suboptimal result. It is even possible that optical illusions are created that falsify the visualization. In many scenarios, rendered graphics accurately depict the source data, but human perception introduces additional artifacts or distorts in addition to existing

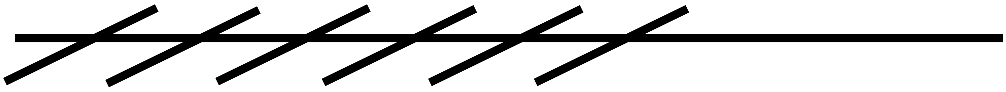


Figure 3.10 — Effect of the tilt illusion. Here, a horizontal line appears to be bent upwards on the left side.

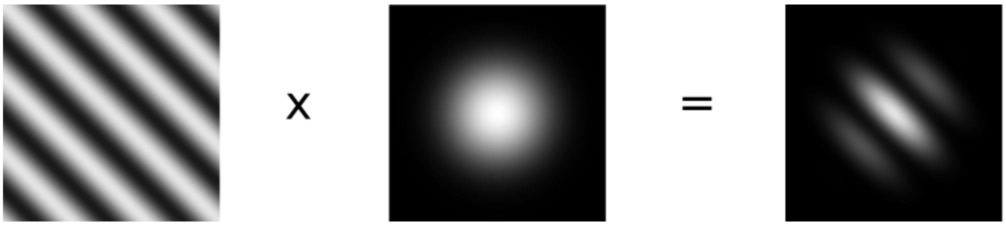


Figure 3.11 — A sinusoidal function is multiplied with a Gaussian function to generate a Gabor filter.

graphical features. One illusion that is very likely to appear in many line-based visualizations is the tilt illusion. It results in the perception of possibly bent lines instead of straight ones, as depicted in Figure 3.10.

In contrast to the method proposed in Section 3.2, the sub-workflow highlighted in blue in Figure I is now fully addressed, since in this section a method for the compensation of the tilt illusion is proposed, or more general the simultaneous orientation contrast. As before, for the quality assessment of advection processes, the frequency spectrum of textures is analyzed. For the approach in this section, the difference lies within how the information gained from the analysis is used. Previously, in Section 3.2, the contributions of frequencies, independent of their directions, were sufficient to compute a key value for a quality assessment. Here, a complementary approach is utilized to perform a compensation of the tilt illusion, thereby, enhancing the visual quality of an image by representing displayed data more truthfully. To this end, perceived directions and frequencies of textures are determined under consideration of the human visual system by utilizing Gabor filters. The proposed compensation model is capable of performing a compensation operating on position-dependent data. Therefore, it can be applied for, e.g., visualizations with superimposed textures.

This might be relevant for visualizations that combine two or more generated images by superimposing them. For example, glyphs could be rendered on top of dense streamline visualizations of vector fields.

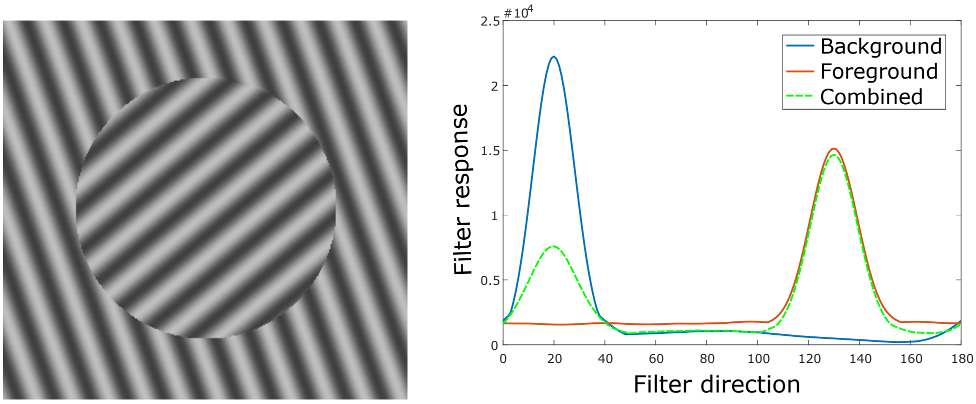
3.3.1 Method

In this section, the origin of used functions and the process of the compensation is described in more detail. The proposed method solely relies on two textures as input: one texture is a designated background texture, while the other one is the foreground texture that is being superimposed in order to provide, e.g., additional information at designated locations. The compensation process consists of three parts: (1) *input texture analysis* to retrieve characteristic values from the textures, (2) computation of the *tilt angle model* that estimates perceived orientation, and (3) the actual compensation of the tilt bias by means of *foreground rotation*.

Input Texture Analysis: In this first step, input textures have to be analyzed to perform the compensation. Perceptually relevant characteristic values are retrieved from the input textures: perceived direction, spatial frequency, and contrast. The contrast of the textures is defined as the difference between the brightest and darkest texture point. Gabor filters are used to retrieve frequency and direction information from the textures. They are generated by multiplying a 2D sinusoidal function with a 2D Gaussian function or a Hamming window (see Figure 3.11). These kinds of filters are commonly used to imitate the frequency and direction perception in the human visual system. Neurons and their associated receptive fields in combination with lateral inhibition of neighboring neurons form a pattern that is similar to that of Gabor filters, which are generated by multiplying a 2D sinusoidal function with a 2D Gaussian function or a Hamming window. These filters are sensitive to the direction and frequency defined by a sinusoidal function.

The computational model performs a convolution of the input texture with these Gabor filters. Filtering results highlight edges that have properties similar to those of the sinusoidal functions of the filters. This is implemented by relying on a filter bank for directions between 0° and 179°, while the frequencies are set to 2^i cycles per degree with $i \in 0, ..., 5$ (assuming a screen resolution of 1920×1200 and a viewing distance of 60 cm). This results in 1080 2D filter responses. Scalar values of each response are aggregated to get a single number that represents the total strength of each filter.

To make these representative scalar values comparable, they are normalized by the size of their corresponding filter. Furthermore, to compute the dominant direction and frequency, previous results are aggregated in two ways: combining all filter responses for each direction and also all values for each frequency. Figure 3.12 shows the result of such a filter process for the first case. Primary perceived directions and frequencies are determined based on the maxima of the normalized responses.



(a) Superimposed images (b) Responses of filters

Figure 3.12 — Result of applying a Gabor filter bank on a superimposed image (a). The plot (b) shows the responses of the filters in tested directions for the foreground, background, and superimposed image. Maxima indicate the main directions.

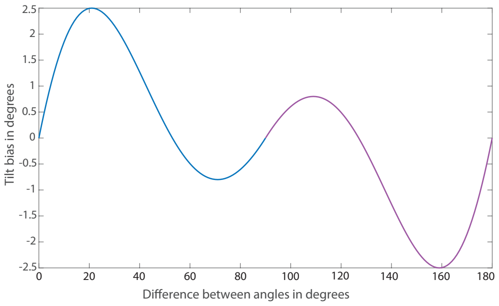
It is important to note that Gabor filtering leads to space-dependent computation of the characteristic values for the texture, i.e., angle, spatial frequency, and contrast are not necessarily constant for the texture but may vary from location to location. While many of the simple, artificial stimuli used in perception research may have constant characteristic values across the whole image, realistic examples from visualization will usually be space-dependent.

Tilt Angle Model: In the second part, these characteristic values are used to estimate the perceived angle tilt at the respective location, i.e., the characteristic values are accessed at the position where the foreground and background textures coincide.

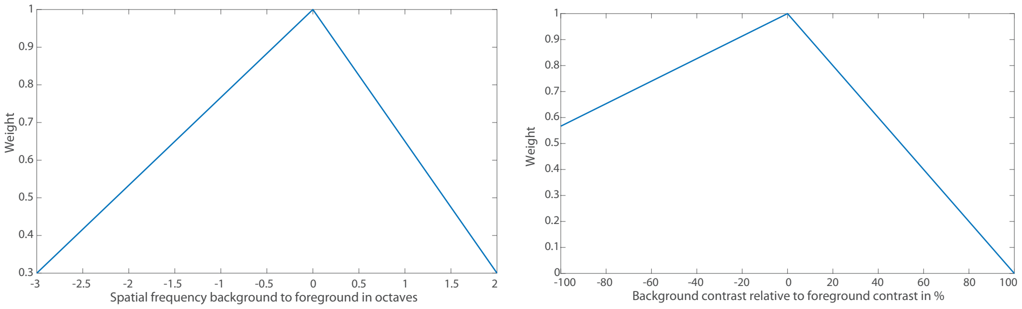
To this end, findings from previous studies and models from vision research are adopted and cast into a computable model. An adjusted tilt bias angle $\tilde{\alpha}_{\text{bias}}$ was determined by multiplying the original tilt bias angle α_{bias} with weights for the influence of the spatial frequency ω_{frq} and the contrast ω_{cont}:

$$\tilde{\alpha}_{\text{bias}} = \alpha_{\text{bias}} \cdot \omega_{\text{frq}} \cdot \omega_{\text{cont}} \tag{3.8}$$

The values of $\tilde{\alpha}_{\text{bias}}$, ω_{frq}, and ω_{cont} are computed by utilizing three functions that are based on previously conducted studies about the effects of the tilt bias concerning differences of directions, frequencies, and contrasts. The first function approximates the original tilt bias angle. The directional bias is defined on the interval [0;180] and is based on experimental results of Schwartz et al. [61].



(a) Directional bias

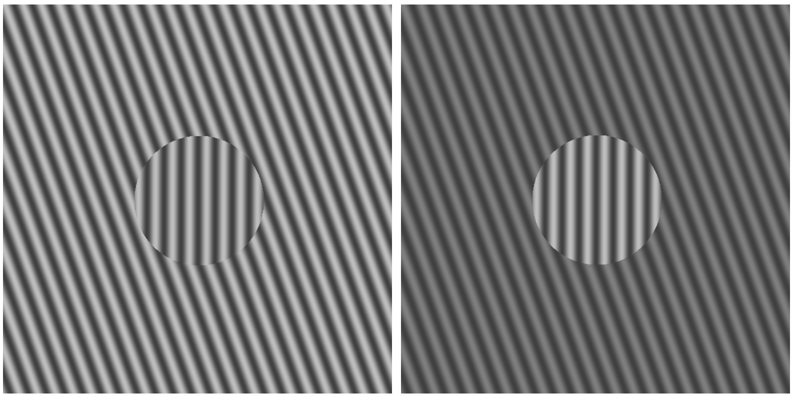


(b) Frequency weight function (c) Contrast weight function

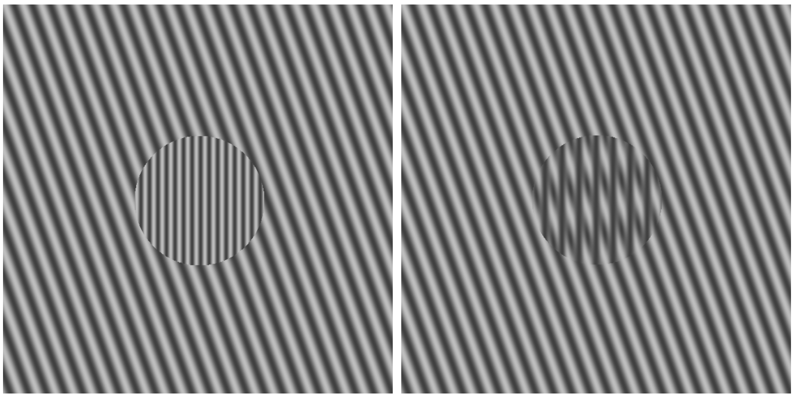
Figure 3.13 — Bias functions that are used during compensation of the perceived direction. (a) Approximated function of the measured tilt bias of Schwartz et al. [61]. The blue part corresponds to positive angular differences in their work, while the magenta colored part represents the negative differences. (b) Function used to determine the weight for the detected spatial frequency of the foreground and the background texture. The weight is one if the spatial frequency has a maximum influence, and zero if not. The curve is based on the results of Georgeson [62]. (c) Function to determine the weight for the contrast difference between the foreground and background texture. The weight is one if the contrast has a maximum influence, and zero if not. The curve is based on the results of Durant et al. [63].

The other two functions were defined in a piecewise fashion and are used to determine weights for the influence of spatial frequency [62] and contrast [63]. Figure 3.13b shows how the influence of frequency is defined on $[-3;0]$ and $[0;2]$ separately. Weights for contrast are available for the input ranges $[-100;0]$ and $[0;100]$ (Figure 3.13c). They indicate that the difference of frequency or contrast of the foreground and background texture influences the effect of tilt bias.

The influence of spatial frequency differences between foreground and background textures has an exponential drop-off in both positive and negative directions. For contrast differences, the dependency of the weight function is



(a) No differences (b) Contrast difference



(c) Frequency difference (d) Transparency

Figure 3.14 — The tilt bias is influenced by the spatial frequency and contrast difference of the foreground and the background texture.

approximately linear for positive and negative values. Note that only Georgeson [62] conducted an experiment to measure the effect of spatial frequency explicitly, while Durant et al. [63] were interested in the after effects of tilt illusions (delayed presentation of background or foreground texture). This also included the simultaneous presentation of both textures resulting in four measurements that were used to approximate the contrast weight function. Effects of tilt bias under different conditions for artificial stimuli are depicted in Figure 3.14.

Foreground Rotation: In the last part, $\tilde{\alpha}_{\text{bias}}$ is finally used to compensate for the tilt bias. The foreground texture is rotated around its center point by the opposite of the adjusted tilt bias angle.

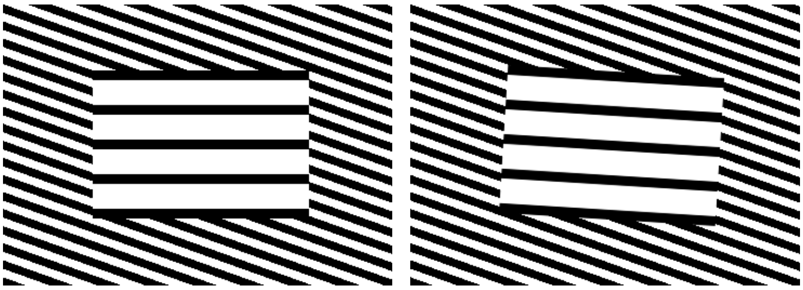
This approach works quite well as long as the foreground object covers a rather small area so that the spatial coincidence between foreground and background can be uniquely determined. Furthermore, a small object may just be rotated as a whole without further considerations to be made.

3.3.2 Results

In this section, the capability of the compensation method is demonstrated. Two scenarios are considered: First, *artificial stimuli* that are commonly used in psychophysical experiments to study the tilt bias or effects related to it. The second scenario shows results of a compensation for an example of *multifield flow visualization* in scientific visualization.

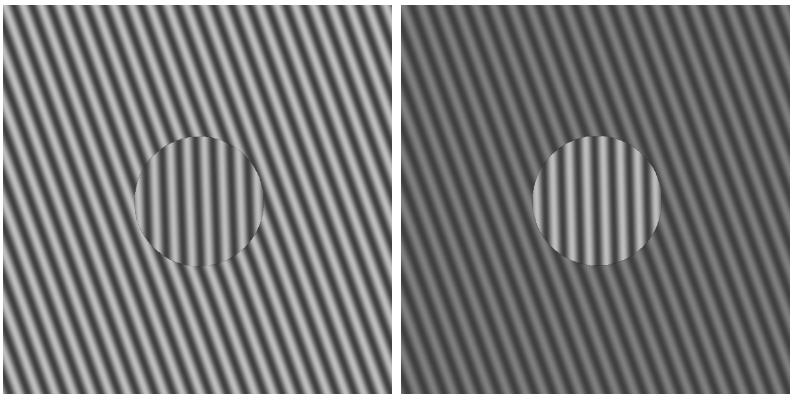
Artificial Stimuli: During the development and refinement of the compensation method, a series of test images were used to rate the compensation qualitatively. One series of test images contained alternating black and white bars, where both foreground and background texture had a rectangular shape (see Figure 3.15). Corners and edges of the foreground textures might be useful for the visual system to perform some autonomous compensation. Hence, circular foreground textures were used in other tests. Additionally, sinusoidal functions were employed to generate striped textures. Related work often uses such kinds of stimuli in the context of measuring the tilt bias.

Through several trial-and-error cycles, problems with the preliminary compensation model were detected and addressed. The filter responses are one example. At first, they were noisy, and it was not possible to detect a maximum since the sizes of filters were not considered. Therefore, normalization of filter responses was introduced. Initial tests with textures of the same contrast and spatial frequency yielded good results. However, as Figure 3.15 shows, the method produced an overcompensation when the spatial frequency between foreground and background was changed. As described in Section 3.3.1, weighting the compensation by the differences in spatial frequency addressed this issue. For the same reasons, a weight for the contrast difference was introduced. These countermeasures prevent overcompensation. Results of successful compensations for simple cases are depicted in Figure 3.16. The test stimuli used so far are based on frequencies and local contrasts that are constant across the image. To further demonstrate the effects of tilt bias and the results of the compensation method, test patterns were created that contain a broad range of frequency, direction, and also local contrast changes by using a Jähne test pattern [71]. In such images, the spatial frequency increases from the center until reaching a defined maximum at the border. They contain frequencies in

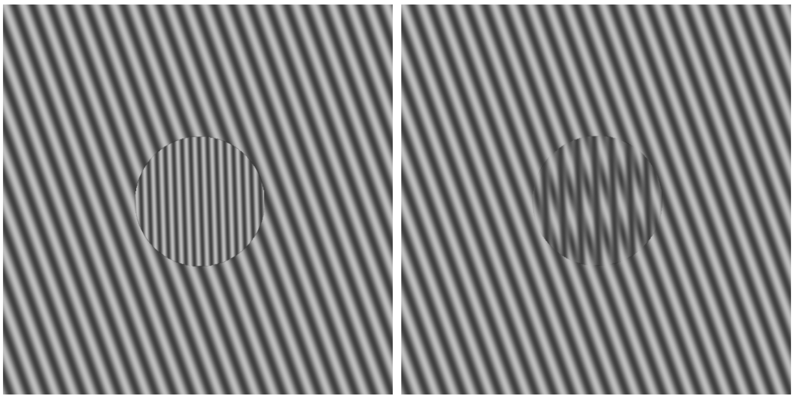


(a) Original image (b) Compensated image

Figure 3.15 — Tilt bias is not present in the original image (a). Using only the directional bias function results in an overcompensation (b).

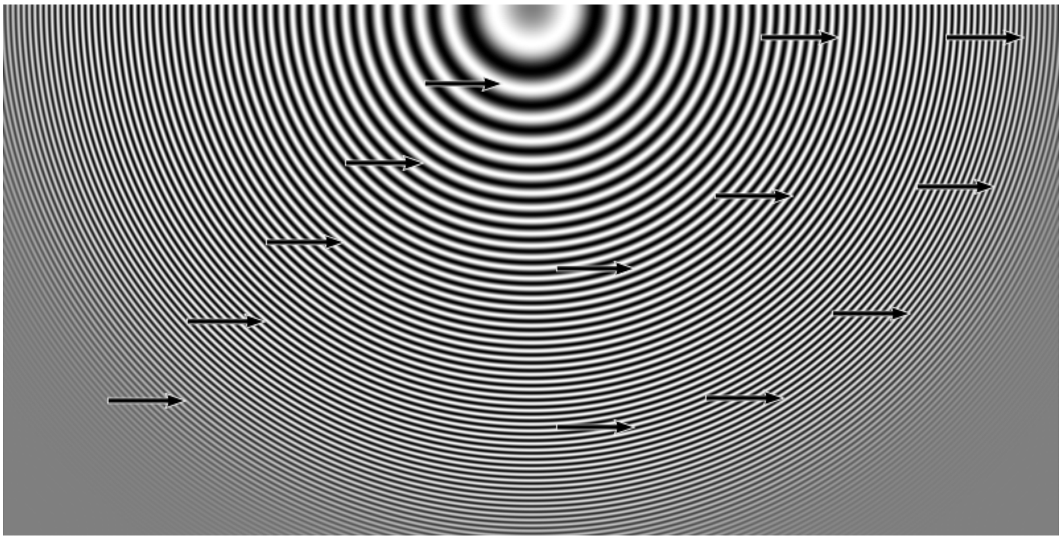


(a) No differences (b) Contrast difference

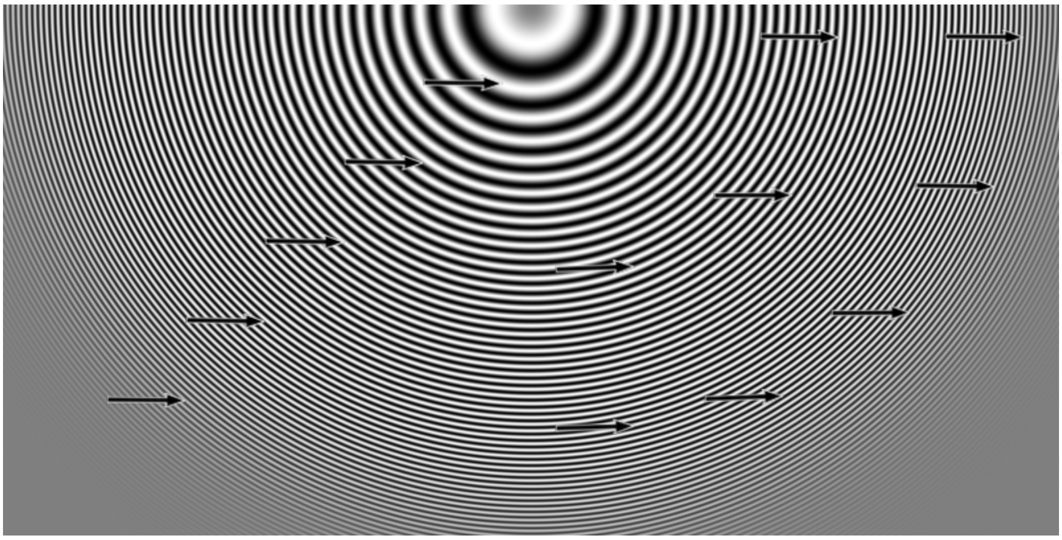


(c) Frequency difference (d) Transparency

Figure 3.16 — Stimuli shown in Figure 3.14 after compensation. In (a) and (d), the compensation was not affected by the contrast and frequency weight. In (b) and (c), the weights reduced the amount of compensation.



(a) Without compensation



(b) With compensation

Figure 3.17 — Arrows are superimposed on a Jähne test pattern. On the left side, in images (a) and (b), arrows are placed on top of the background test pattern on positions with varying frequencies and a constant direction. On the right side of both images, arrows are arranged in two circles, leading to a constant frequency and a varying direction. In a viewing distance of about 20 cm, some arrows seems to be tilt in (a). This effect is compensated in (b).

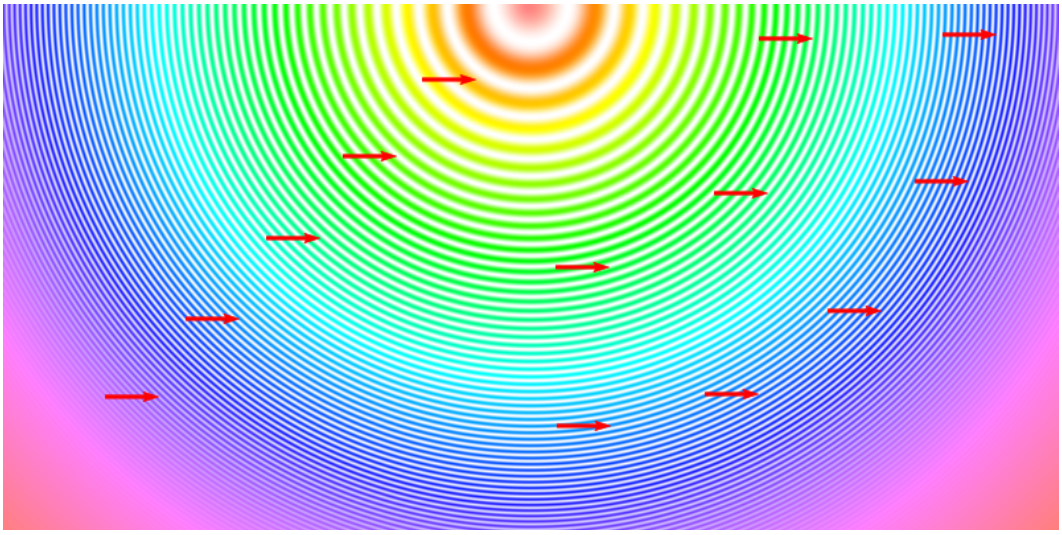


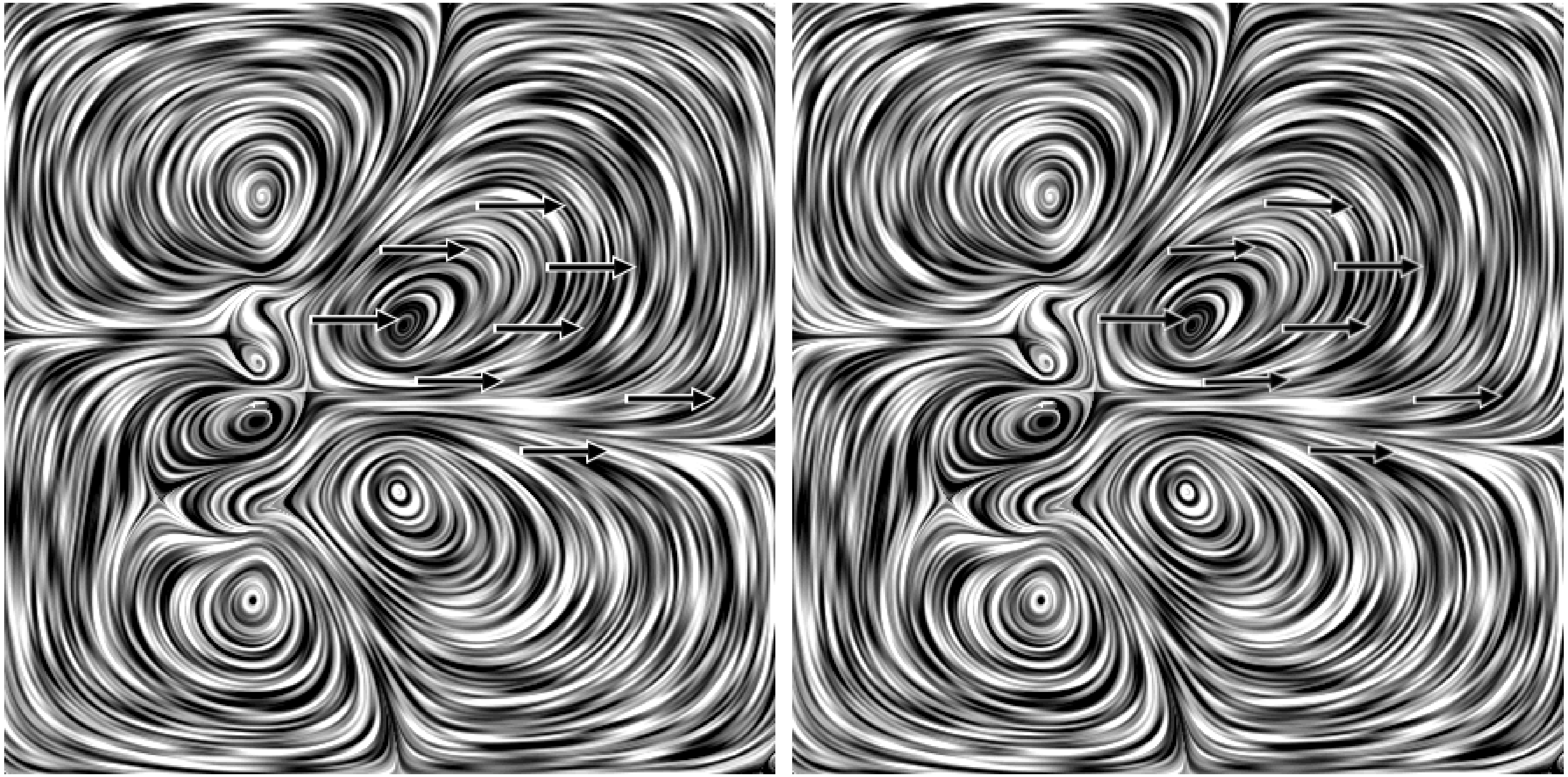
Figure 3.18 — Arrows are superimposed on a Jähne test pattern. On the left side, for each arrow position, the frequency is changing, and the direction of the background texture is constant. On the right side, arrows are arranged in two circles leading to a constant frequency and a varying direction. Here, color is used to show the influence of color contrast on the tilt effect, which is less present compared to Figure 3.17.

all directions and are generated using the function

$$g(x) = g_0 \sin\left(\frac{k_m |x|^2}{2r_m}\right)\left[\frac{1}{2}\tanh\left(\frac{r_m - |x|}{\omega}\right) + \frac{1}{2}\right]. \tag{3.9}$$

Here, x is the vector offset from the image center, $|x|$ the distance from the image center, r_m is the maximum radius of the pattern in the image, and k_m specifies the maximum instantaneous frequency. Furthermore, tanh can be seen as an approximation to a step function, where r_m is the location of the step and ω is the width of the transition. The following parameters were used to generate the test patterns: $k_m = \frac{\pi}{2}$, $r_m = 400$, and $\omega = 40$.

Figures 3.17 and 3.18 only use the lower half of the Jähne pattern, due to its symmetry. Arrows were placed on a radial line to demonstrate the effects of varying frequency and constant perceived direction of the background texture. Furthermore, by arranging arrows on circular paths, the impact of changing the perceived direction of the background texture and a constant spatial frequency is shown. Figure 3.17a shows the original image where some arrows seem to be rotated by several degrees. This effect is compensated in Figure 3.17b.



(a) Image without compensation

(b) Image with compensation

Figure 3.19 — Arrows are superimposed on background textures, which depict LIC visualizations of a vector field. The three arrows at that bottom of (a) are affected by the tilt illusion in a viewing distance of about 20 cm. In the compensated image (b), they appear again straight.

Multifield Flow Visualization: So far, the compensation method was demonstrated only for artificial stimuli. Therefore, it is of importance to show that the method can also be applied to realistic scenarios. Here, foreground textures might be glyphs or other small graphical elements that are superimposed over a background texture to provide additional localized information.

A specific, yet typical example from scientific flow visualization is discussed: the simultaneous plot of two 2D vector fields in a single image. One of the vector fields may be visualized by arrow glyphs that yield a coarse representation. A complementary visualization is chosen for the other vector field and shown as background texture. Here, a dense texture-based method is used in the form of line integral convolution (LIC). This method allows us to visualize fine details of vector fields that might be hard to detect in the coarse grids of glyphs.

In such an example use case, the simultaneous orientation contrast between the LIC background and the glyph foreground can result in a tilt illusion. Misperceptions in the orientation of superimposed arrows can then lead to false readings of flow direction. Therefore, compensation is crucial for a correct interpretation of the vector fields. Figure 3.19a shows an example without compensation. Here, an analyst might have selected a region that is of interest and manually placed arrows. For this example, all glyphs are assumed to be perfectly horizontal. Tilt bias from orientation contrast creates the appearance of the lower three arrows going in slightly different directions, while the upper five are not affected. After compensation, all glyphs appear horizontal in Figure 3.19b. The proposed method works on each arrow individually, i.e., the glyphs are rotated against the tilt angle derived at the respective location on the LIC background.

3.3.3 Discussion

Although tests using different artificial stimuli show promising results and it was demonstrated that the method can be applied in more realistic scenarios, there are still a couple of topics that should be investigated, included, or tested to improve the compensation further and to test its validity.

The first topic that should be investigated in more detail concerns the weight functions. The retrieved function for the frequency weight seems to be reliable since it is based on the results of an experiment that specifically tested the effects of the frequency of the tilt bias. This is not the case for the contrast weight. Here, only four measurements of an experiment were used that tested the after effects of tilt illusions in the case of a simultaneous appearance of foreground and background texture. In both cases, it would be beneficial to verify the results of the frequency experiment and to improve the contrast weight function by conducting experiments with a more appropriate sampling of the parameter space. Another aspect that is not addressed in the here proposed method is the impact of color on the tilt illusion. This was subject to previous research [72, 73, 74] where measurements were taken concerning color signals (L, M, S-cone signals derived from RGB images) and luminance. Here, color contrast can reduce the tilt effect as shown in Figure 3.18. Note that this image was not compensated, but the tilt effect is lessened compared to Figure 3.17a. In the best case scenario, Equation 3.8 could simply be extended by additional weight terms related to the influence of color. In the worst case, there are dependencies between color, contrast, and frequency that cannot be modeled as simple as proposed here.

The next topic that could be of relevance is to incorporate Gestalt laws into the compensation process. Here, e.g., horizontally aligned arrows on a virtual horizontal line are less influenced by the tilt bias according to initial tests and observations. Humans, or their visual system, seems to have certain expectations or makes assumptions in the presence of some geometric structures.

The last topic is related to the estimation of the dominant frequency and direction. As described in Section 3.3.1, the maximum response for direction and frequency is currently utilized, but as depicted in Figure 3.12b, it is possible that there are several distinct local maxima, which could be considered during the compensation.

Chapter

4

Visualization Techniques for Eye-Tracking Data

For eye-tracking data, there are many sophisticated evaluation methods, as described in Section 2.5, but still, some basic problems are not addressed appropriately. Therefore, this chapter[1] focuses on the three aspects related to eye-tracking data and their analysis.

The first topic addressed in this chapter is the annotation of data. Currently, no system focuses on the annotation of single fixations of a scanpath in ambiguous situations. Most of the existing systems and methods target the definition of AOIs to perform labeling of fixation data, which is the basis for many techniques like sequence or semantic analysis. Although automatic approaches are faster in terms of assigning labels after AOIs are defined, it is not a reliable process.

---

[1] **Parts of this chapter have been published in:**
R. Netzel, M. Burch, and D. Weiskopf. Interactive scanpath-oriented annotation of fixations. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ETRA '16, pages 183–187. ACM, 2016 [3], © 2016 ACM.
R. Netzel and D. Weiskopf. Hilbert attention maps for visualizing spatiotemporal gaze data. In *IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)*, pages 21–25, 2016 [2], © 2016 IEEE.
A. Kumar, R. Netzel, M. Burch, D. Weiskopf, and K. Mueller. Multi-similarity matrices of eye movement data. In *2016 IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)*, pages 26–30, 2016 [4], © 2016 IEEE.
A. Kumar, R. Netzel, M. Burch, D. Weiskopf, and K. Mueller. Visual multi-metric grouping of eye-tracking data. *Journal of Eye Movement Research*, 10(5), 2018 [5], © Journal of Eye Movement Research. DOI: 10.16910/jemr.10.5.11. License: CC BY 4.0

Additionally, the assignment of AOIs is, depending on the type of stimulus, a tedious task. In ambiguous situations of overlapping or neighboring AOIs, the assignment is no longer certain. The proposed framework fills this gap by providing intuitive methods for the visual and interactive annotation of fixations for both static and dynamic stimuli without the need to define AOIs. In contrast to the proposed framework, there is an approach that supports the automatic annotation of fixations based on speech recognition and linguistic processing of voice recordings of a conduced thinking-aloud study [75]. Nevertheless, the annotations need to be checked, since this approach is still error-prone.

The second aspect is the spatiotemporal visualization of eye-tracking data. Here, *Hilbert attention maps* are introduced, which show the spatial and temporal positions of gazes simultaneously in a 2D visualization for both static and dynamic stimuli, i.e., stimuli that may carry relevant spatiotemporal characteristics. In contrast to other approaches, *Hilbert attention maps* do not rely on the definition of AOIs. Instead, a regular grid is employed on the stimulus. The grid resolution can be adjusted to control the level of spatial aggregation. A space-filling Hilbert curve is associated with this grid, leading to a projection from 2D space to a 1D curve that respects local neighborhoods.

The last part addresses eye-tracking metrics, or in general, metrics used for evaluation. They are essential to many evaluations and associated techniques. The question is now which metrics should be used and which are irrelevant or convey the same kind of information, and are therefore redundant. Unfortunately, there is no designated set of metrics that can always be used, since the relevance of metrics can change depending on the stimulus and task. Here, this problem is addressed by utilizing a workflow that contains analysis methods to explore possible metrics and identify groups of participants exhibiting similar characteristics.

4.1 Related Work

For data analysis, different aspects are of importance depending on the data and used tasks. For example, the analysis and visualization of eye-tracking data can be classified into three categories: First, data annotation (Section 4.1.1), since many methods rely on annotated data to analyze or interpret the data based on additional semantic information provided by the annotation. Second, there is the topic of spatiotemporal representation of eye-tracking data (Section 4.1.2), which is important, since analysts are interested in both spatial- and temporal patterns. Finally, the analysis of eye-tracking data is often based on some metrics, resulting in an analysis and visualization of multivariate data

(Section 4.1.3). In the following sections, previous work concerning these three topics is described.

4.1.1 Data Annotation

Some tools address the issue of data annotation by assigning a label to a scene, or an AOI contained within a scene. These frameworks can be categorized into different groups: frameworks for the annotation of static stimuli (images), those for dynamic stimuli (videos), and the ones that utilize crowd-sourcing support.

For video data, Benayoun et al. [76] proposed an authoring tool that utilizes shot detection to extract semantic links between objects within shots. The objects and links are then labeled and verified by the analyst. The post-production framework by Bertolino [77] performs a semi-automated spatiotemporal segmentation. The segmentation-based method by Zhi and Jie [78] only requires the analyst to mark the desired object that is then tracked frame-by-frame. While most snapshot detection methods fail for long video sequences, the technique by Ring and Kokaram [79] overcomes this problem: the analyst selects the mask of an object that is used to steer the semi-automatic detection. Other semi-automatic annotation systems are *ViPER* [80] and *ANVIL* [81]: Rectangular AOIs can be annotated in keyframes, intermediate positions are obtained through interpolation, and AOI positions can be adjusted manually.

Crowd-sourcing is an effective means of annotating many data elements. The *Open Video Annotation Project* [82] allows analysts to specify time intervals within a video, attach labels, and have a look at other annotations from different users. In a long-term study, Vondrick et al. [83] used their software to annotate a large number of video data by object tracking; if the tracking is no longer accurate, an update is performed by specifying the correct position of the object. For static images, there are much fewer implementations that support the annotation process. One example is the work by Deng et al. [84], where a crowd-sourced image annotation pipeline is introduced that utilizes *Amazon Mechanical Turk (AMT)* to validate image class candidates.

There are also methods that utilize machine learning to determine objects that were being watched automatically. Haji Mirza et al. [85] train a classifier on the trajectory of the eye movements. As another example, Fathi et al. [86] use semi-supervised learning in combination with segmentation to extract objects from videos. However, a machine-based analysis needs to be trained or tuned for specific stimuli and tasks. Therefore, it cannot be used as a generic and ready-to-use approach in ambiguous annotation scenarios.

With commercial software of eye-tracking vendors—such as Tobii Studio 3.0—it is also possible to define AOIs on static and dynamic stimuli. However, the annotation of single fixations within the context of their scanpath is not facilitated.

In general, there is much research on scanpath representation, comparison, and visualization. Goldberg and Helfman [87] discuss different scanpath representations that can provide additional insight into scanning strategies. Raschke et al. [88] describe a visual approach to the comparison of scanpaths. However, these techniques are not designed to support labeling specifically.

4.1.2 Spatiotemporal Representation

As discussed in the survey by Blascheck et al. [41], spatiotemporal data visualization plays an important role in eye-tracking analysis, since the spatial and temporal positions of gazes are of relevance for both static and dynamic stimuli.

To visualize spatiotemporal eye-tracking data, the use of animation is one option, for example, in the form of animated bee swarms or animated attention heatmaps [89]. While animations can be intuitively understood, because they are congruent with the representation of the data, their support of data analysis tasks is dubious [90], as exemplified for animated scatterplots [91].

The most natural representation would be a 3D space-time cube to show two spatial and one temporal dimension [92]. However, such 3D visualization suffers from occlusion, imperfect depth perception, and difficult data selection in volumes. It is possible to place several small multiples for different time spans side-by-side [93] to overcome these problems. However, small multiples do not scale well for many time spans and do not directly support visual comparison between different times; therefore, it is hard to identify temporal patterns. The motion-compensated heatmap [92] produces a static attention map for dynamic stimuli but lacks the visualization of the temporal information of gazes. Finally, there are function plots that either show the x- or y-component of gazes over time, which obviously lack information about the 2D spatial configuration.

The definition of areas of interest (AOIs) is one way to add further information that can be utilized to improve temporal analysis by essentially reducing the 2D space of gaze positions to a list of AOIs, leading to a considerable *compression* of the data. Sequences of AOIs can then be visualized [94], gaze frequencies displayed in a Sankey diagram [95], and hierarchies of AOIs can be utilized for analysis on different aggregation levels [96]. The general idea of parallel scanpath plots [97] was proposed to visualize transitions between manually designated AOIs. There, the AOIs are represented at discrete positions on the x-axis of a plot, whereas the y-axis indicates when an AOI was fixated, thus visualizing the temporal order of fixations. Unfortunately, it is quite hard and time-consuming to define AOIs, in particular, for dynamic stimuli. Nevertheless, they are used in many fields for evaluation of spatiotemporal eye-tracking data, e.g., for the analysis of web search tasks [98] or source code debugging [99]. Holmquist et al. [36] discuss the concepts of AOI-based evaluation in more detail.

Another possible way to visualize spatiotemporal eye-tracking data is to project the 3D data to a lower dimensional domain, which could be achieved by, e.g., using space-filling curves (SFCs) [100, 101]. SFCs were previously utilized in visualization literature, e.g., to aid in computing a layout for treemaps [102] or preserve locality in the projection from 3D to 1D [103]. Other pixel-oriented visualization techniques use a Hilbert curve for a mapping from 1D to 2D. Here, the components of a 1D data vector are mapped on the Hilbert curve of a 2D image [104, 105].

4.1.3 Multivariate Eye-Tracking Data

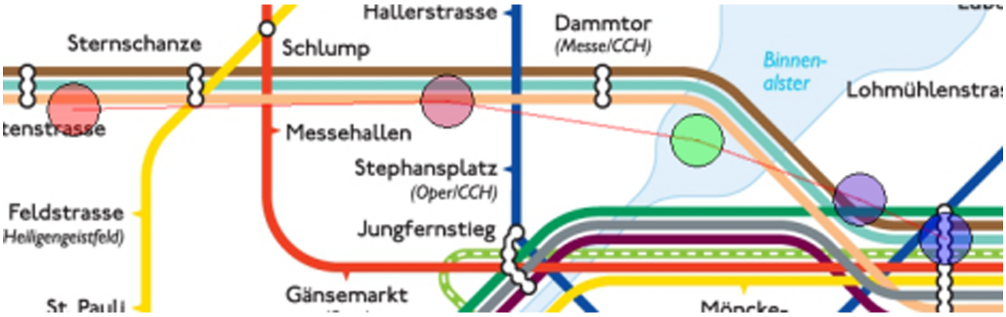
Understanding participants' behavioral patterns and learning about commonalities and diversities in these, is a challenging task. These types of analyses have been done in the past for many fields, such as usability research using machine learning, data mining [106], grouping of users for music recommendation [107] based on complex amino acid sequences [108]. Such applications also exist in the field of eye tracking. In this context, the results of a conducted user study are presented in Chapter 7 with respect to task performance. Furthermore, participants were grouped on the basis of a fixation label sequence of scanpaths and the bimodality coefficient. To group participants, Burch et al. [109] performed an evaluation of tree diagrams concerning task performance and recorded eye-tracking data. Kurzhals et al. [110] grouped users on the basis of similarity functions such as the Levenshtein distance applied to scanpaths, a function based on attention distribution, and one based on AOI transitions [111]. Anderson et al. [112] also conducted scanpath comparison in an experiment to reveal the similarities within and between the scanpaths of individuals while looking at natural scenes.

West et al. [113] used fixation sequences to highlight groups or clusters of sequences that are similar. Typically, traditional approaches for comparing eye-movement behavior of several study participants focus on the scanpaths by aligning them as well as possible to detect the commonalities [97, 114]. All of these applications have shown that it is important to group users on the basis of their common behavior. However, the above-mentioned grouping approaches are only based on few standard descriptors of eye-tracking data utilizing, e.g., fixation duration, saccades, AOIs, or other metrics (see Section 2.2.2) related to scanpaths. Although grouping subjects using clustering based on such derived values is possible, the quality of the obtained groups is not always optimal. From a machine learning perspective, the result of the grouping or clustering highly depends on the used feature vector that contains attributes that describe characteristics of data instances and their relation. In the context of eye tracking, these attributes are metric values derived from recorded eye-tracking data. The main challenge lies now in the selection of appropriate metrics, which is not easy to achieve since the characteristics that the metrics should describe highly depend on the used stimuli and task. Hence, there is no fixed set of metrics that can be used to achieve a grouping of participants. Therefore, it is necessary to investigate and select the space of metrics for each new experiment and type of stimulus.

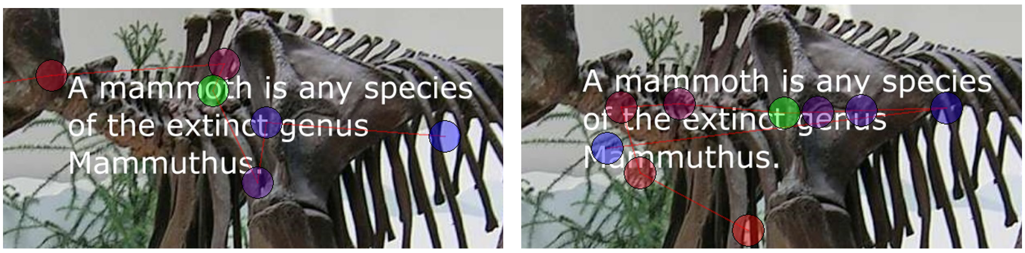
4.2 Annotation of Eye-Tracking Data

This section describes a strategy for enriching eye-tracking data by a semantic annotation that can be applied to static (images) as well as dynamic (videos) stimuli. The method only requires sequences of fixations as input data, along with the associated stimuli. Such data is provided as an export by any of the typical software shipped with eye-tracking devices. Data may be available for one or several participants. The problem setting that is addressed here is as follows: how can fixations be annotated efficiently in a flexible way by taking into account the scanpath context of the fixations (i.e., the previous and subsequent fixations)?

The concept of using AOIs is avoided here, since that would implicitly lead to annotations of fixations but would also ignore the scanpath context. Figure 4.1a illustrates an example in which AOI-based annotation fails. The example is taken from an eye-tracking experiment in which participants were asked to follow metro lines. This experiment is discussed in Chapter 7 of this thesis. In Figure 4.1a, the participant is following a metro line, but one fixation (marked green) is (slightly) off the line that the participant was most likely following; yet, an annotation should be assigned that describes the intention of following the metro line. Another example is shown in Figures 4.1b and 4.1c. The center fixation (green) covers multiple AOIs and therefore the intention of the subject is not clear. The scanpath reveals that the subject was performing different actions: inspecting the skeleton and reading the text. Put differently, the context of the previous and subsequent fixations is indispensable for *cleaning up* annotations that would otherwise be assigned inaccurately—here, due to inaccuracies from



(a) Following a line



(b) Looking at mammoth skeleton (c) Reading text

Figure 4.1 — Gaze plots of short scanpath segments. Fixations in (a) before and after the green fixation are at visual structures (metro lines) of the stimulus (© Robin Woods, licensed from Communicarta Ltd) and, thus, could be annotated according to an AOI (of the metro lines). However, the green fixation cannot be labeled as *following the metro line* without taking into account preceding and succeeding fixations. The scanpaths in (b) and (c) show different patterns at the same area of the image (© Lou Gruber) covering two overlaid AOIs (skeleton and text), and therefore indicate two different behaviors: inspecting the skeleton and reading the text.

eye tracking or AOI definition. Even without inaccuracies, there are cases where the scanpath context is relevant for assigning appropriate labels since intentions could only be inferred from that context.

The proposed concept is implemented in a lightweight annotation framework that facilitates efficient manual annotation. It allows an annotator to work through a sequence of fixations while visualizing the context of the scanpath in the form of previous and subsequent fixations. The visualization is laid over the stimulus in the background to provide the stimulus context. This is supported for both static and dynamic stimuli. Optionally, automatic initial labeling according to areas of interest (AOIs) is possible. The design of the system is flexible and versatile: it is easily configurable, in particular, it supports user-defined annotation schemes. Through import and export of data files, the system can be smoothly integrated into typical existing workflows of eye-tracking experiments and accompanying data evaluation.

The here proposed framework supports an early step in data analysis, namely the data annotation (fixations of eye-tracking data), which is also covered by the sub-workflow colored in red in Figure I.

The main benefit of a visually supported fixation annotation lies in scenarios with *difficult* stimuli, e.g., real-world stimuli, unconstrained experiments, or studies *in the wild*. Such scenarios frequently lead to ambiguities that cannot be easily resolved by fully automatic annotation through AOI-based labels.

4.2.1 Design Goals and Rationale

Two problems cannot be addressed by AOI-based annotation alone. First, there might be ambiguities caused by AOIs that are small, very close together or overlap; there might be ambiguities in assigning the finite fixation area (representing the size of the foveal region and inaccuracies of the eye-tracking device) to a single AOI. Second, the semantics of the cognitive processing associated with a fixation may depend on the context of the scanpath. This problem is illustrated for the example of reading a metro map (Section 4.2.3): even if a fixation on a metro station was accurately recognized, the semantics of the fixation might have different meanings (such as following one line versus transit to another).

To resolve these problems, an annotation is necessary that is guided by what is typically facilitated by AOIs, but that is more flexible and robust. The driving factor for the initial development of the system goes back to the needs for annotation for a metro map study, see Figure 4.1a and Section 4.2.3. Literature research and survey of existing tools revealed that there was a lack of annotation software that would meet required needs. The developed system is flexible enough to cover a wide range of applications that go beyond the example of the metro map study.

For a structured design of an appropriate system, the following requirements are identified:

Flexible access to the fixations to be annotated: The annotator should have random access to fixations (e.g., to correct specific annotations), but should also be able to work through fixation sequences systematically.

Visualization of the scanpath and stimulus context: The subsequence of fixations before and after the current fixation should be shown, overlaid on top of the stimulus. This avoids drawing the whole scanpath, which would lead to visual clutter.

Configurable and extensible annotation scheme: There should be support for a configurable list of labels to guarantee consistent labeling. There should be a way of adding previously unaccounted labels to the scheme.

Support for static and dynamic stimuli.

Support for AOI-based automatic annotation (optional): Fixations should be automatically assigned labels through correspondence to AOIs if such AOIs are available. Misannotations can be corrected with subsequent manual annotation.

Smooth integration in existing workflows: Existing eye-tracking or statistics software should not be replaced. Instead, the system should be able to import (unlabeled) eye-tracking data and export labeled data for further statistical analysis or visualization.

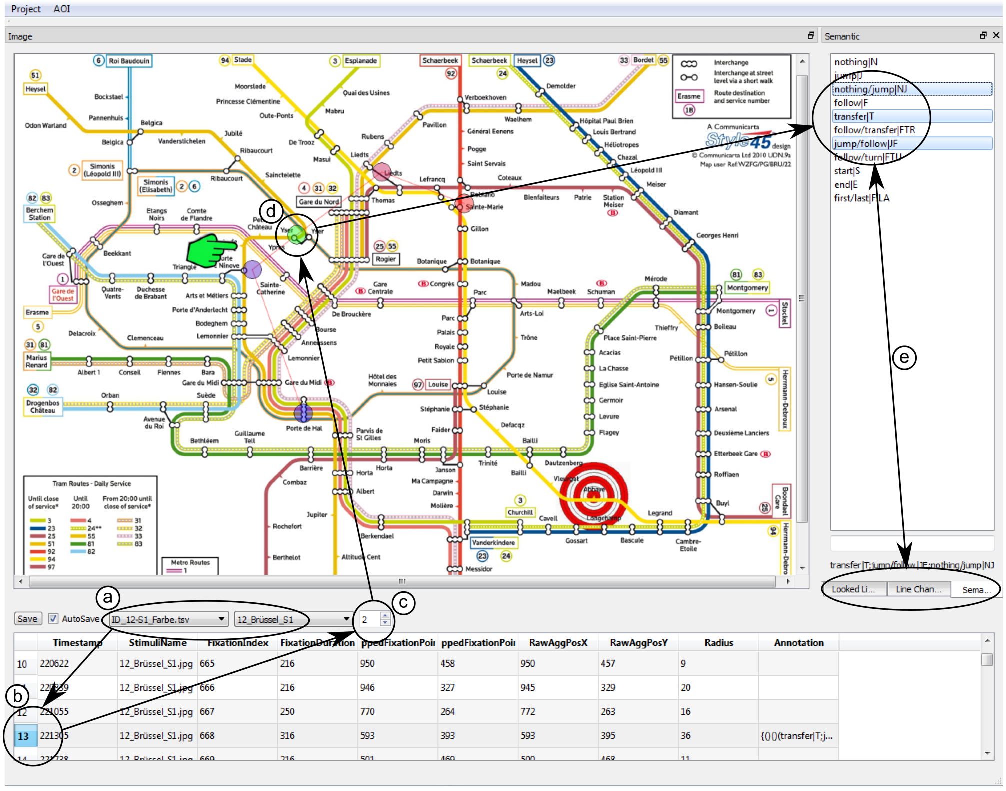
Although a manual annotation is rather time-consuming, in some cases, it is necessary, or helpful, to handle the annotation of ambiguities by including user input. Here, an initial AOI-based annotation could be used as a starting point. Afterward, problematic fixations can be determined automatically by utilizing an appropriate metric. Such a metric could be, e.g., the number of AOIs that are in proximity, the actual distance to the nearest AOI border, or the number of AOIs that are assigned to the fixation, in case of nested AOIs. Typically, there are only a few fixations that require a manual annotation utilizing the context of the scanpath.

4.2.2 Components

The framework depicted in Figure 4.2 consists of three visual components: data view, stimulus view, and label selection view. Additional non-visual functionality complements these.

Data View: It allows the annotator to manage the data. The annotator can select a scanpath of a participant and stimulus. The fixation sequence is then displayed in a tabular fashion. There are four mandatory fields: timestamp, stimulus name, and x- and y-coordinates of the displayed fixations. The program automatically adds an annotation column to store assigned labels. Other data is optional information that could support the annotation process.

Stimulus View: It shows the subsequence of fixations on top of the stimulus. The data view is linked with the stimulus view: A row (fixation) is shown once it is selected. Besides, many fixations that precede and succeed the selected fixation are displayed. A circle represents a fixation. Its size is derived from



☑ AutoSave ID_12-S1_Farbe.tsv 12_Brüssel_S1 2

nothing|N
jump|J
nothing/jump|NJ
follow|F
transfer|T
follow/transfer|FTR
jump/follow|JF
follow/turn|FTU
start|S
end|E
first/last|FLA

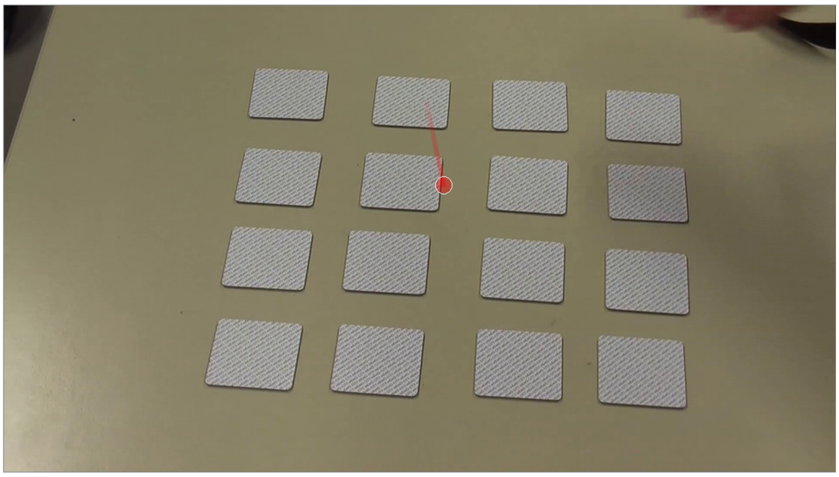
transfer|T;jump/follow|JF;nothing/jump|NJ

	Timestamp	StimuliName	FixationIndex	FixationDuration	ppedFixationPoi	ppedFixationPoi	RawAggPosX	RawAggPosY	Radius	Annotation
10	220622	12_Brüssel_S1.jpg	665	216	950	458	950	457	9	
11	220839	12_Brüssel_S1.jpg	666	216	946	327	945	329	20	
12	221055	12_Brüssel_S1.jpg	667	250	770	264	772	263	16	
13	221305	12_Brüssel_S1.jpg	668	316	593	393	593	395	36	{()()(transfer|T;j...

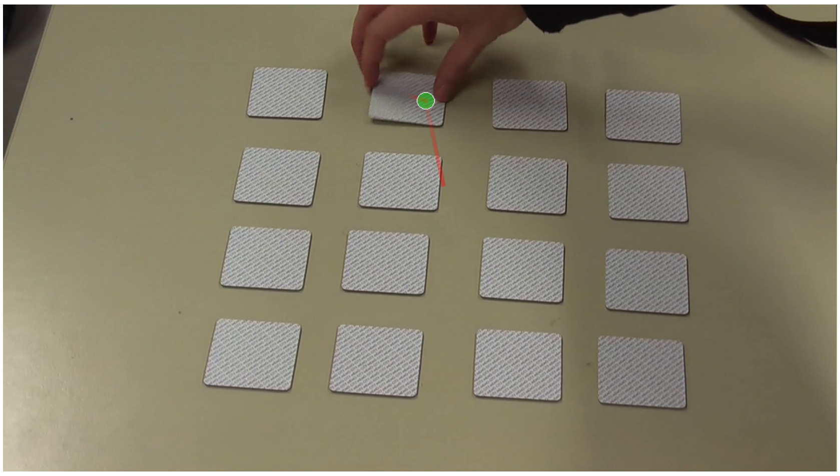
Figure 4.2 — Snapshot of the system with a sequence of actions necessary for annotation: (a) Selection of a participant and stimulus (in the data view). (b) Selection of a fixation in the table (scanpath). (c) The length of the shown sub-scanpath can be specified with a spinbox. (d) The stimulus view marks the center of the sub-scanpath; the center fixation that is currently being annotated is colored green. (e) Label selection view: three labels (highlighted in blue) of the category *Semantic* (abbreviated as *Sema...* at the bottom) are selected. Metro map © Robin Woods, licensed from Communicarta Ltd.

parameters of the experimental setup to mark the area on the screen covered by the fovea in the human eye. The color of the fixation encodes the temporal order of the sub-scanpath (with a linear red/blue gradient). The selected center fixation is highlighted in green color. In the case of static stimuli, the selected sub-scanpath is readily visualized as a gaze plot. For dynamic stimuli, the fixations of the sub-scanpath are not within one frame but spread over several frames.

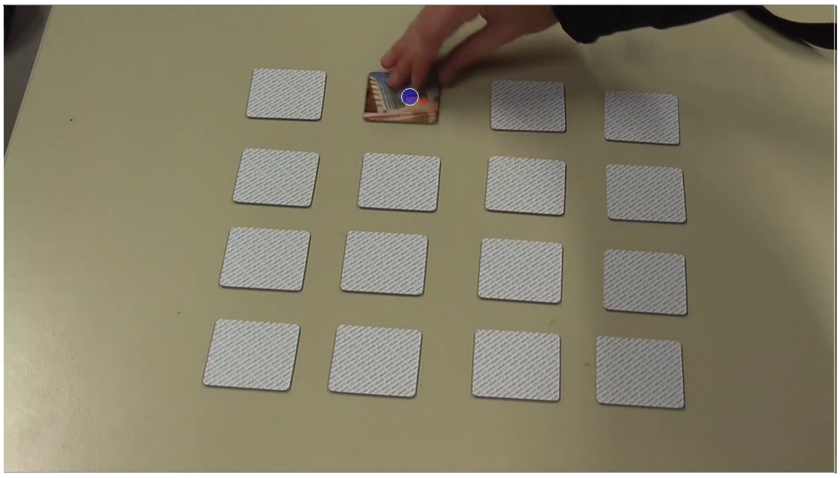
Here, the frames of the video are extracted that contain fixations of the sub-scanpath. The fixations are then positioned in their associated frames, aligned horizontally in temporal order. Tapered links are used to include directional information across frames, see Figure 4.3.



(a) t-1



(b) t



(c) t+1

Figure 4.3 — Annotation for dynamic stimuli. Frames that contain fixations are presented in temporal order. Fixations are colored according to their temporal order with a red/blue gradient. The center fixation (at timestep t) that is currently being annotated is highlighted in green. Tapered links are attached to each fixation to visualize direction information.

Label Selection View: The system is capable of handling multiple categories that can be annotated for each fixation. Each category can contain multiple labels. After a fixation is selected in the data view, appropriate labels for each category can be chosen. If labels were assigned previously, these labels will be pre-selected. The labels are saved in a format specified by the annotator within a project file. The categories and their labels are also contained within the project file. If a label is not yet available, the annotator can define a new label and assign it to a fixation.

Automatic Processing: An optional automatic annotation using AOIs is available. The label of an AOI defines the initial label of the fixations contained in this AOI.

Data Import/Export: Eye-tracking data is imported from tab-separated text files, which is a common file format of most eye-tracking software. Output files adopt the same format: they contain all input columns and an additional column with the assigned labels.

4.2.3 Use Case

The software was in productive use for 140 hours by a research assistant who performed the annotations. 39,404 fixations were annotated, yielding an average processing speed of about five fixations per minute. This time includes the selection of a fixation that should be labeled, the investigation of the corresponding sub-scanpath, the assignment of appropriate labels, and possible corrections of misannotations. The sub-scanpath had a length of 5 to provide enough scanpath context.

Multiple fixation labels from the following three categories were used: the metro lines that the participant was looking at (30 labels), the possible transitions to other lines (30 labels), and finally the possible semantic (12 labels).

The semantic labels were used to annotate actions a participant performed during the execution of the task or to mark special regions that participants had looked at (like the start and destination stations). By defining specific labels, it is possible to deal with possible errors in gaze position introduced by inaccuracies of the eye-tracking device; one example of such inaccuracies was discussed for Figure 4.1a.

Based on this semantic annotation of fixation data, an evaluation of a study on the readability of metro maps was performed. See Chapter 7 for a detailed description of the study and its results.

4.3 Space-Time Visualization of Eye-Tracking Data

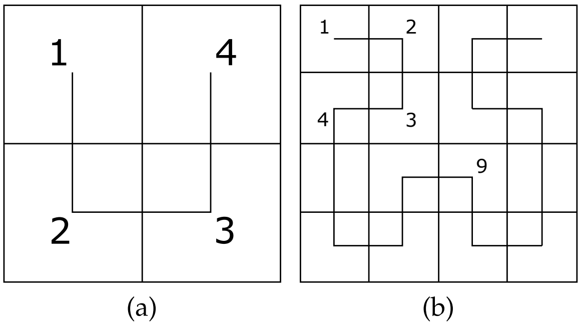
Aggregated views of data are important to identify common patterns of reading behavior in eye-tracking experiments. For example, attention heatmaps are used to show the spatial distribution of gaze or fixation data in a compact and easy-to-understand way. Here, hotspots on a stimulus are often associated with centers of attention of participants. These hotspots often correlate with features within a stimulus that are of general interest or that are relevant to solve a given task. A general problem of aggregated views is that some important information might be lost unless additional precautions like, e.g., saving provenance information, are taken. In particular, traditional attention heatmaps suffer from a complete loss of temporal information.

However, the order of fixated locations is often important during the analysis. Even (relative) time points can be relevant, e.g., to identify attentional synchrony between participants. To overcome this problem *Hilbert attention maps* are introduced, which provide a 2D static visualization of the spatiotemporal distribution of gaze points. The basic idea is to reduce the 2D spatial domain of gaze locations to a 1D domain. Then, the other dimension of the diagram can be used to show the temporal information associated with gazes. The projection from 2D space to 1D is based on a space-filling Hilbert curve. This visualization combines the best of traditional attention maps (i.e., the display of spatial patterns) and temporal plots (i.e., the display of temporal order and positions). Spatiotemporal Hilbert attention maps have some advantages: they do not require any definition of areas of interest (AOIs) or any other time-consuming preprocessing including data annotation as discussed in Section 4.2.

Hilbert attention maps provide a static visualization of temporal information that lends itself to visual data analysis, and they work for static and dynamic stimuli. With this approach, an analyst can get insights into spatiotemporal eye-gaze patterns of participants. Since it is meant to be a static visualization, there are currently no sophisticated interactions implemented, therefore, the sub-workflow colored in red in Figure I is not fully addressed.

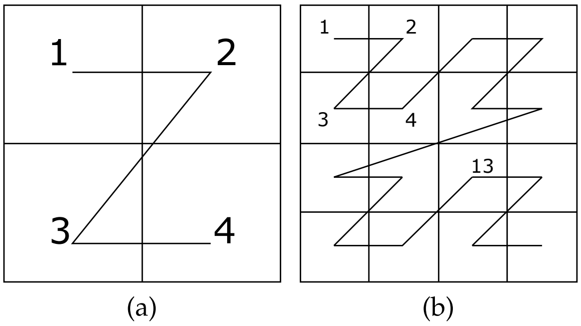
4.3.1 Space-Filling Curves

An elegant method to project an 2- or 3-dimensional area to 1D are space-filling curves (SFCs). Here, each point or cell of the area is visited exactly once, thereby, indexing the whole area. The idea is to utilize a basic curve for a recursive refinement of the area as depicted in Figure 4.4 for a Hilbert curve, which preserves locality [115] quite well, compared to other space-filling curves, as the Z-curve illustrated in Figure 4.5. Here, long jumps are visible between



(a) (b)

Figure 4.4 — Examples of Hilbert curves for aggregation level one (a) and two (b). The numbers indicate the order in which the curve traverses the grid, thereby defining the index number of the grid cells.



(a) (b)

Figure 4.5 — Examples of Z curves for aggregation level one (a) and two (b). The numbers indicate the order in which the curve traverses the grid, similar to Figure 4.4.

some visited sectors. Utilizing space-filling curves will lead in general to the separations of areas in proximity and, therefore, a design goal is to minimize the introduced distortion. For a Hilbert curve, a U-shape is used as basic curve element (respectively a Z-shape for Z-curves). In a recursive refinement step, each element of a curve that covers a cell will be replaced by a possibly rotated basic curve element.

4.3.2 Visualization Method

The presented approach allows relating temporal and spatial information in a static 2D visualization—the Hilbert attention map. This is achieved by a projection of the 2D continuous spatial domain onto a discrete Hilbert SFC. First, the *level* of the curve is specified, which implicitly defines a regular grid of size of $2^{level} \times 2^{level}$ that covers the whole currently investigated stimulus.

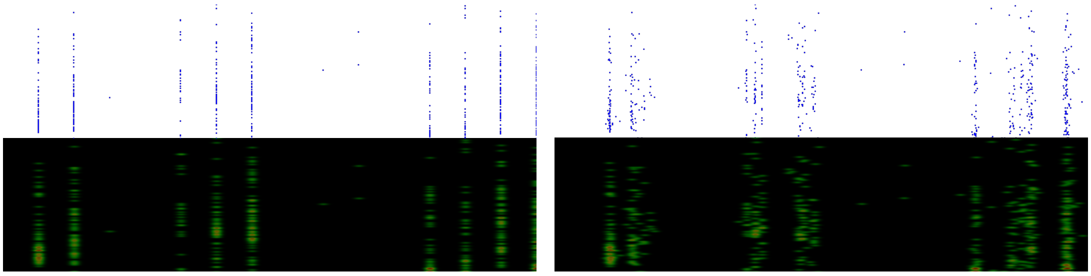

(a) level 2 (b) level 8

Figure 4.6 — Influence of the Hilbert curve level on the extent of aggregation. The upper images show dot displays, the bottom images depict heatmaps. Data of 20 participants are used to generate the plots.

The Hilbert curve can be recursively defined (see [100, 101]) to produce curves of increasing resolution, and Figure 4.4 illustrates such a recursion step. Each grid cell is assigned an index depending on its position on the Hilbert curve, and all gaze data within one cell inherit the same index. This cell membership is the mechanism to aggregate in a spatial neighborhood.

Now, each cell can be addressed with one coordinate, which serves as the x-axis of the spatiotemporal diagrams. The y-axis is used to represent time. By increasing the *level*, the extent of aggregation is decreased, which leads to more fine-grained visible patterns. The maximum *level* depends on the resolution of a stimulus. For the data, with a resolution of 1920 × 1200 pixels, the visualization started changing only slightly for *levels* greater than eight. Therefore, for these examples, the *level* is set to eight. Figure 4.6 illustrates the impact of the aggregation levels.

Two variants of visualizing the 2D Hilbert attention map can be recommended. The first variant uses dots that are placed directly at the x- and y-positions that come from the construction of the SFC, see top rows in Figure 4.6. Such a dot display was shown to be more user-efficient than alternative displays of scatterplot data [116]. For this application, the dot display is particularly useful when a visualization *close-to-the-raw data* is needed and outliers have to be shown. The second variant produces a heatmap that shows frequencies in the Hilbert attention map by color coding, see bottom rows in Figure 4.6. In these examples, a kernel density estimation is used with a Gaussian kernel with standard deviations $\sigma_y = 1$ and $\sigma_x = 5$ (guided by the temporal sampling rate and size of the foveal area). The heatmap visualization is best for showing main trends or patterns in the data.

In general, the Hilbert curve preserves locality, but the preservation is not perfect.

Depending on the *level*, some points that are in proximity to the 2D spatial domain could end up on far apart positions on the Hilbert curve. Furthermore, it is hard to relate patterns in the Hilbert attention map to positions on the 2D stimulus. To address both problems, brushing-and-linking [117] is supported between the Hilbert attention map and traditional 2D spatial visualizations such as gaze plots.

The presented approach could be extended by, e.g., depicting sub-scanpaths based on the selected gaze data for several participants in a separate view. In this view, each scanpath could be adjusted in length to investigate where participants were looking before and afterward in the spatial domain. One could also think of the opposite direction: selection of areas in the 2D spatial domain and highlighting the respective gaze data within the Hilbert attention map. These kinds of interaction methods could mitigate the problem that Hilbert curves do not preserve locality perfectly since it would be possible to inspect selected data in the 2D spatial domain.

4.3.3 Examples

Hilbert attention maps are produced for benchmark datasets [118] that contain recorded gaze data of participants watching videos with carefully chosen content, which should trigger eye movements exhibiting intended characteristics.

The first video shows a moving red car. The participants were asked to follow the car, which results in smooth pursuit. The camera was moving as well. In the second example, participants watched a person playing a memory game. Here, the participants were asked to look at a covered card of a pair when the other card of this pair is uncovered. With this spatial memory of participants could be tested. This time, the camera was stationary.

In Figure 4.7 keyframes of a video are depicted that are associated with the three parts of the video: the car is entering the scene (Figure 4.7 ⓐ), performing a turn (Figures 4.7 ⓑ and ⓒ), and leaving the scene (Figure 4.7 ⓓ). The slightly tilted movement of the car in the first and last part is indicated by a linear (straight) pattern of gazes in the Hilbert attention map, whereas the gaze data of the middle part exhibit a curved or s-shaped pattern. The heatmap shows especially well that these regions attracted the most attention, i.e., attentional synchrony. In general, synchrony can be identified in the attention map by single hotspots at a time slice (constant y-position in the plot). Figure 4.7 also shows that smooth pursuit can be detected: curve patterns that are not at a constant x-position in the plot. This example also illustrates a limitation of the projection from the 2D spatial domain to an SFC: some points that are close in 2D space might become separated through the projection (see the split

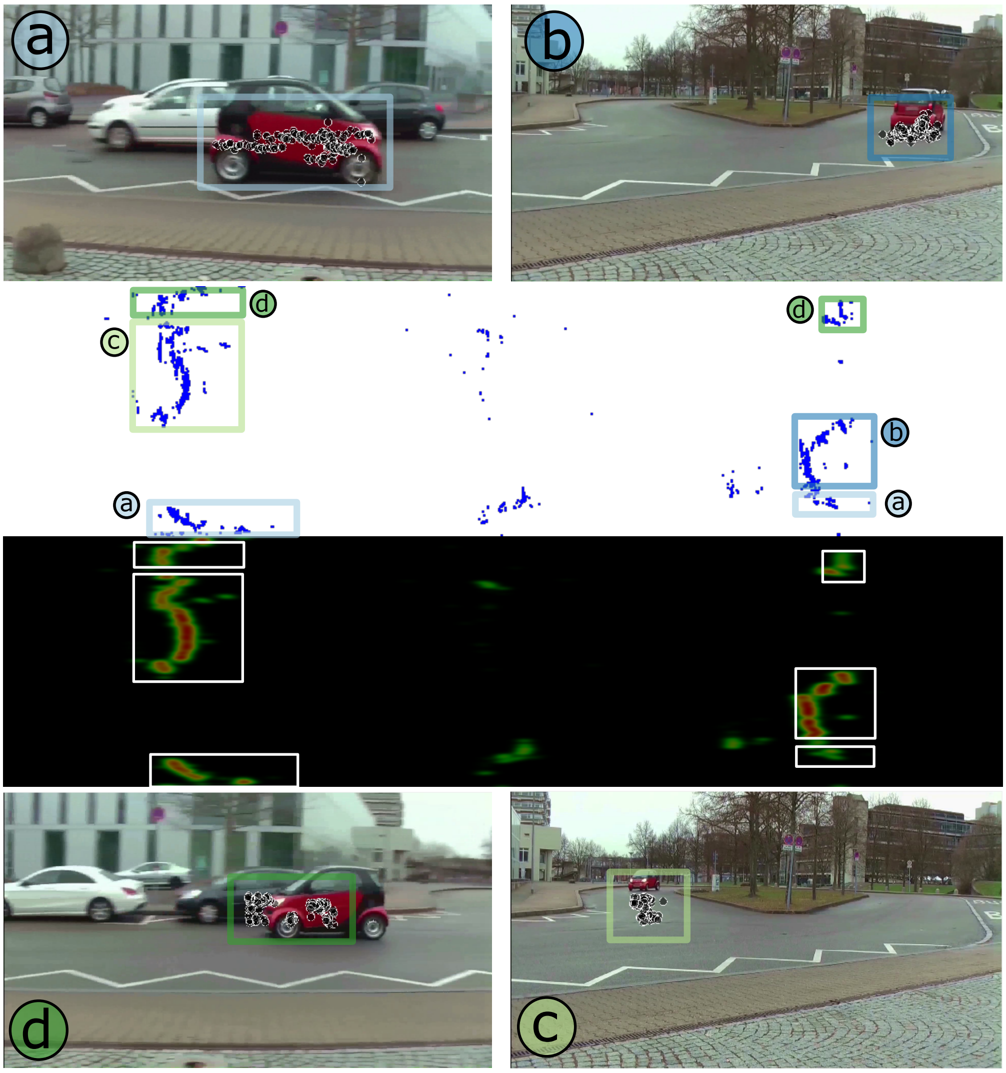


Figure 4.7 — Visual patterns caused by smooth pursuit and attentional synchrony between three participants following the red car shown in a video. Images **a** to **d** (subsequent frames of a video) contain the gaze data (black dots) and stimulus of the central frame of the respective time intervals. The different phases (**a** to **d**) are selected by boxes in the dot display of the Hilbert attention map (second row). The heatmap (third row) depicts the gaze density.

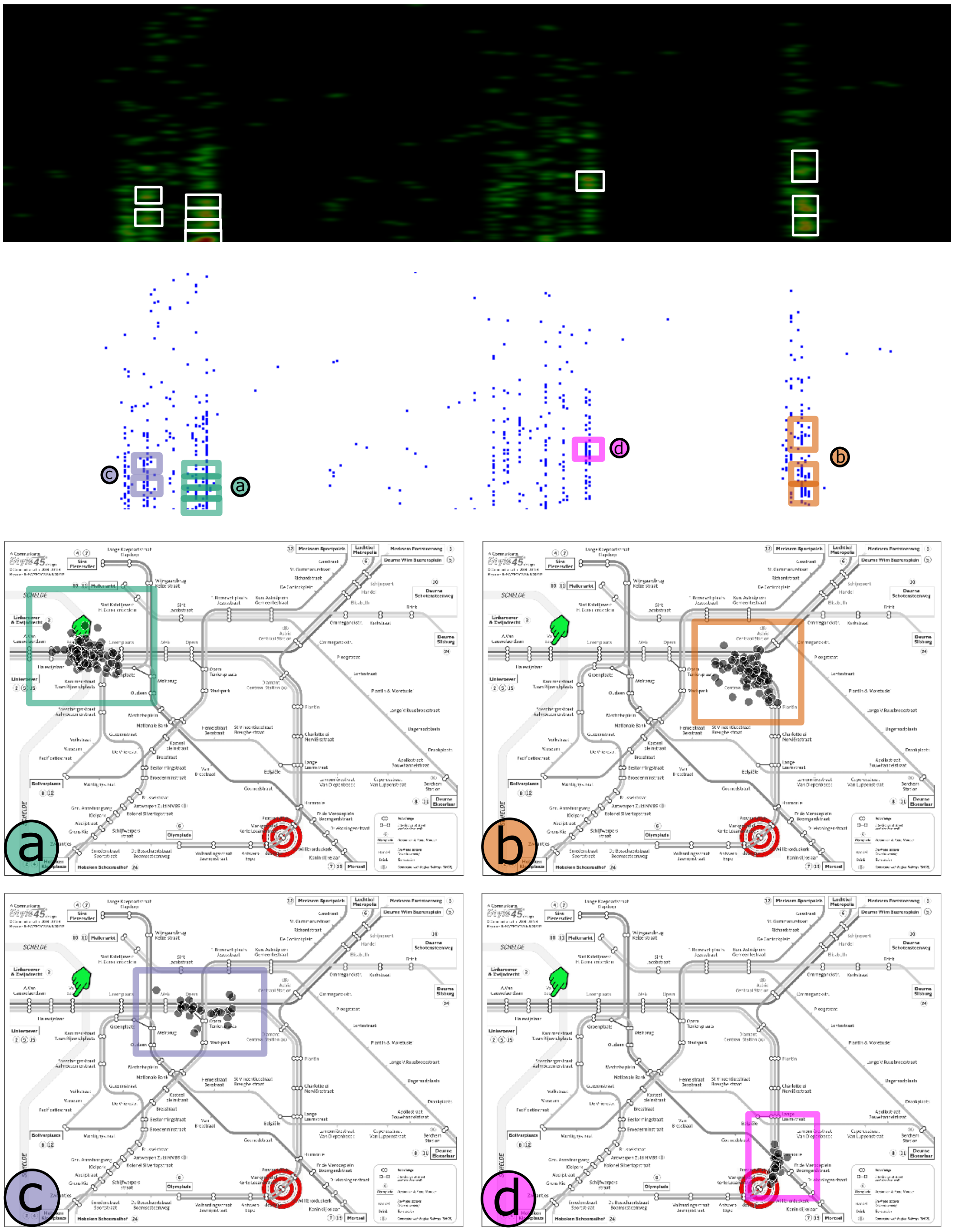


Figure 4.8 — Visualization of recorded attentional asynchronous gaze data of 40 participants solving a path-tracing task with a metro map. The images at the top show the dot display and heatmap of the complete time span. The bottom part focuses on interesting subareas with associated gaze data.

boxes for phases ⓐ and ⓓ. However, clear patterns can still be seen, and brushing-and-linking allows an analyst to understand their spatial positions on the stimulus.

Figure 4.8 shows another example. This time it is based on asynchronous gaze data. The time participants spent on fulfilling the task varied between 5.74 s and 21.9 s. First, the overview of the complete spatiotemporal domain (top images) is investigated. Here, one can identify some interesting hotspots. More importantly, it is possible to derive a temporal sequence of inspected areas and, thereby, find spatiotemporal transition patterns that are of interest. In this example, in the beginning, the attention of participants was shifting between the starting area ⓐ and a larger junction ⓑ, which is located half-way to the destination. Intermediate inspected areas ⓒ indicate a forward search strategy. Furthermore, there are signs for a backward search strategy starting at the destination ⓓ and moving to the larger junction ⓑ.

4.4 Utilization of Multiple Eye-Tracking Metrics

Another aspect that is often of relevance for analysts is to extract groups of participants that exhibit a similar characteristic, according to some defined criteria. For this purpose machine learning and data-mining approaches are often used to generate such groupings automatically. The used criteria are encoded within a feature vector that characterizes data instances. Generating a sufficient feature vector is in general not an easy task, since it requires domain knowledge and a good intuition about the data that are used for training, testing, and also for the data that are supposed to be classified or grouped. Even if a good feature vector was constructed, it not easy to trust the model that is using this feature vector for classification [13]. Training algorithms often used as a black box to generate a model that disguises the importance of each component of the feature vector and, therefore, on which basis groups are formed. This is, even more, the case for boosting algorithms that generate a strong and descriptive feature vector based on an automatic selection and weighting of weaker features.

This problem is also present in eye tracking. Here, a feature usually consists of metric values derived from recorded eye-tracking data to characterize the behavior of participants and ultimately to group them. However, the selection of appropriate metrics depends on the task as well as the used stimuli and is, therefore, often not simple. It requires a good intuition and understanding of possible outcomes of an experiment.

If only one metric is used, a common approach includes the creation of a

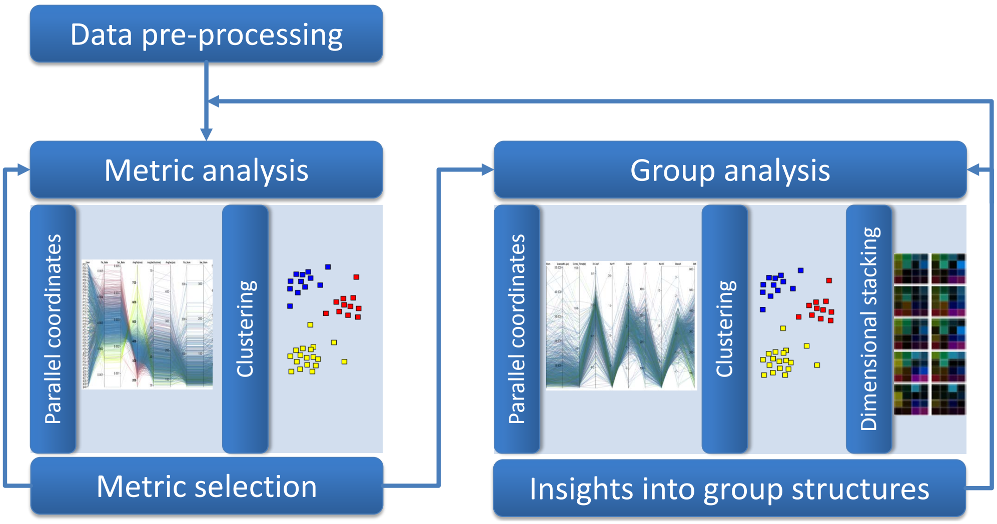
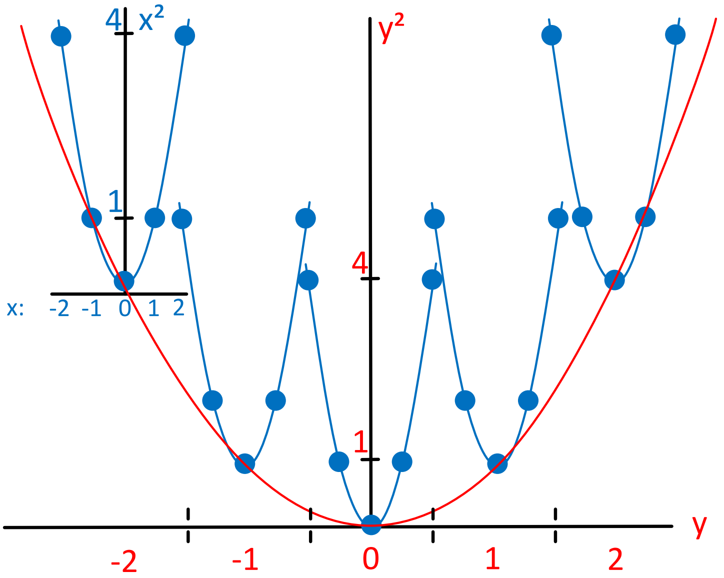


Figure 4.9 — Overview of the established workflow: Pre-processed data is fed into the *Metric analysis* where metrics are selected, based on given methods. These can be used either in the *Group analysis* or in a different evaluation that is taking place concurrently. Insights gained from the *Group analysis* can be used to refine the *Metric selection*. In both analysis blocks, parallel coordinates and clustering can be used. Additionally, the *Group analysis* supports dimensionally stacked similarity metrics of participants.

similarity matrix, but with an increasing number of metrics, it gets harder to establish correspondences between several matrices and to draw conclusions from them. Therefore, an approach to visualize multivariate data is here described based on dimensional stacking of similarity matrices. Furthermore, a possible way to analyze the used eye-tracking metrics is proposed, which utilizes parallel coordinates. The combination of these two approaches allows one to overcome the problem of a suitable selection of metrics and provides insights into the formation of groups and their intergroup similarity.

The workflow defined in Figure 4.9 illustrates the different steps for the analysis of eye-tracking data, starting with a data pre-processing and metric computation, followed by a cycle of the *Metric analysis* using parallel coordinates and clustering, and a participant *Group analysis* based on similarity matrices. The eye-tracking metrics described in Section 2.2.2 are used. This workflow represents a specialized version of the general workflow highlighted in red in Figure I and also includes interactions to investigate the metric space using parallel coordinates to identify behavioral patterns of groups of participants. This is also the subject of Hilbert attention maps proposed in Section 4.3, but



(a)

Figure 4.10 — Stacking of dimensions for two dimensions $x, y \in \{-2; -1; 0; 1; 2\}$ to visualize the functions $f(x,y) = x^2 + y^2$. Five buckets are created for y. In each y-bucked, the x-dimension is embedded.

Hilbert attention maps directly use recorded eye-gaze data and do not rely on interaction.

4.4.1 Dimensional Stacking

Exploring multi-dimensional data is often not an easy task, especially if more than three dimensions are involved. With up to three dimensions, humans can relate to data, since humans live in and perceive a 3D world. For more dimensions, it gets increasingly harder to imagine how data values are distributed or functions look like. Methods were introduced that project the data from a higher-dimensional space into a lower one to reduce the complexity to a level which humans are accustomed to. Such projections often introduce distortions of distances of data points, if the intrinsic dimensionality of the data is higher than the dimensionality of the projected space. The concept of dimensional stacking has been introduced in the past [119, 120, 121] to address this issue: dimensions are embedded within dimensions. This is illustrated in Figure 4.10 for 1D on the example of the simple function $f(x,y) = x^2 + y^2$. Here, the 2D function is decomposed into 1D functions of x and y, where the function of y is used in the first step resulting in the red parabola. For each value of $y \in \{-2; -1; 0; 1; 2\}$, a container is established that is used to display

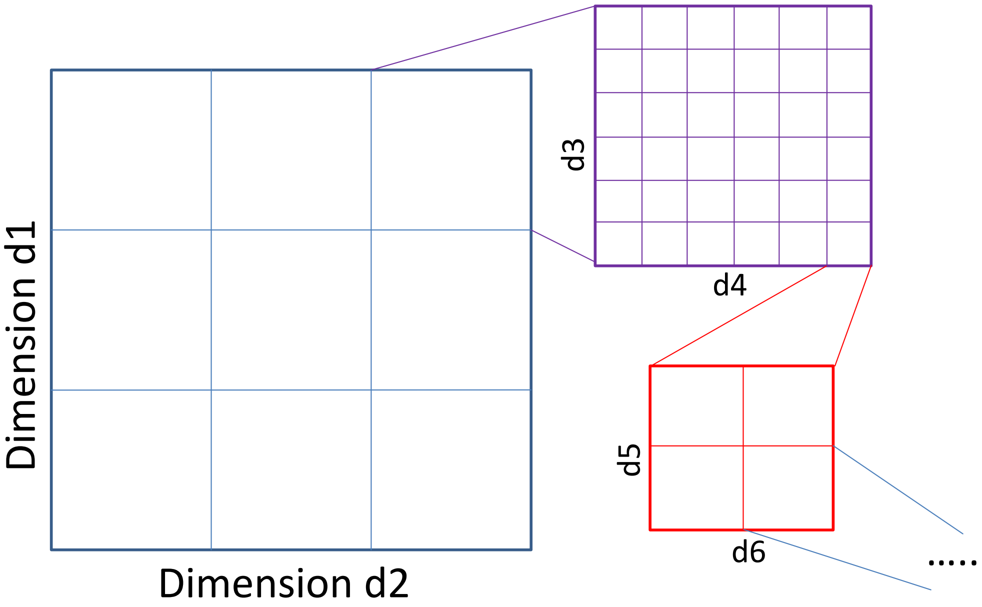


Figure 4.11 — N dimensions are stacked in 2D by selecting two dimensions, create buckets, and embed the next dimensions in each bucket. This can be applied recursively.

the next dimension with a vertical offset defined by the y-function for every value of y. This leads to multiple parabolas highlighted in blue. Here, values $x \in \{-2; -1; 0; 1; 2\}$ are used to visualize the x-function. The same method can be utilized to generate 2D dimensionally stacked visualizations. Here, in each recursive step the two axes are used as x- and y-axis (see Figure 4.11). This can be expanded to 3D.

4.4.2 Similarity Matrix Visualization

A matrix-based representation is one of the basic ways of visualizing 2D data. However, it is difficult to visualize multi-dimensional data using a matrix. Therefore, the following describes an approach to overcome this problem utilizing dimensional stacking. For generating the dimensionally stacked similarity matrix (DSSM), participants P_i with $i \in \{1, ..., p\}$ and all stimuli S_l with $l \in \{1, ..., s\}$ are considered to calculate all metric values M_k with $k \in \{1, ..., m\}$. For each metric M_k, a similarity matrix is calculated with similarity values sv_{i,j,M_k} for each pair of participants $P_{i,j}$ with $i, j \in \{1, ..., p\}$.

The similarity values are calculated using the Euclidean distance (Equation 4.1).

$$sv_{i,j,M_k} = \; || \; M_{k,P_i} - M_{k,P_j} \; ||_2 \tag{4.1}$$

All matrices are now combined into one $p \times p$ matrix, where each cell contains m similarity values sv. This corresponds to the dimensional stacking. However, it is not easy to arrange the m metrics per cell to be visually appealing and still be able to recognize groups of participants using the combined matrix representation by visual inspection. Therefore, it is necessary to adapt the order to the metric values for all cells and also the order of the participants on the x- and y- axis.

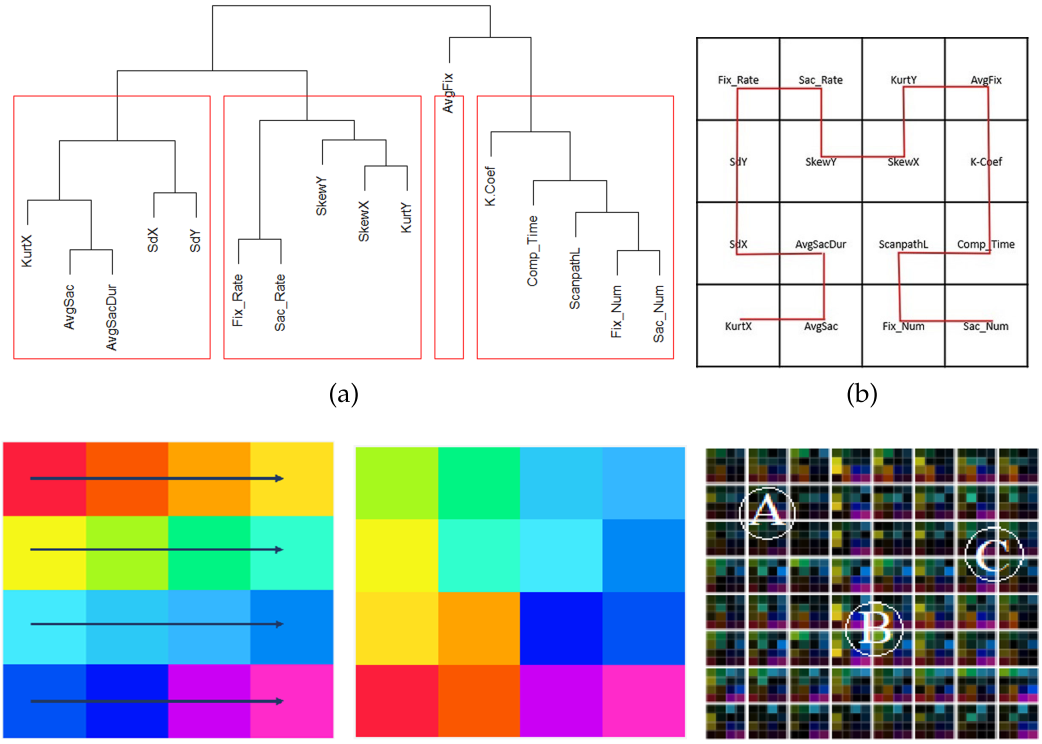
The first part, the reordering of the metric values, is addressed by calculating correlation coefficients between all the pairs of m metrics, which results in $\frac{n!}{2!(n-2)!} = \binom{n}{2}$ values. Based on these correlation coefficients, agglomerative hierarchical clustering is performed to place similar metrics next to each other. A resulting dendrogram is shown in Figure 4.12a. Now, a numeric identifier is assigned to each leaf (metric) of the dendrogram, which is increasing from left to right, starting with one. This number is associated with the position within a metric cell, which is defined by a Hilbert space-filling curve (see Figure 4.12b). In the example in Figure 4.12c, 16 colors with varying hue are chosen from the CIE LAB colorspace to represent the metrics since varying hue helps in grouping categorical data [122]. Colors are assigned to the sub-grid of elements within a cell. A Hilbert space-filling curve again defines the order. Actual values are now encoded in the luminance channel. The result is shown in Figure 4.12d, whereas an example of a final dimensionally stacked similarity matrix for eight participants with 16 metrics, is depicted in Figure 4.12e.

The reordering of the participants on the axes corresponds to the second part. This is necessary to highlight groups of participants in the dimensionally stacked similarity matrix. To achieve this, first an affine combination $\tilde{M}_i = \sum_{k=1}^{m} \omega_k * P_{i,M_k}$ of all metric values M_k is calculated for each participant P_i, with the condition that the sum of weights $\sum \omega_k = 1$. Next, based on these *meta metric values*, agglomerative hierarchical clustering is performed to determine the order of the participants.

4.4.3 Use Case

The main challenge for the clustering is to choose the right combination of metrics and also to select appropriate weights for the computation of *meta metric values*. Therefore, an additional analysis of the derived metric values is necessary. Here, parallel coordinates are used as basis for the metric analysis as indicated by the workflow depicted in Figure 4.9.

The first step in most evaluations is to generate an overview of the data that should be analyzed. Here, the data corresponds to metrics derived from recorded eye-gaze data. This is shown in Figure 4.13 using parallel coordinates.



(a) (b)

(c) (d) (e)

Figure 4.12 — Overview of the procedure followed in the generation of a dimensionally stacked similarity matrix (DSSM). The dendrogram is shown in (a) reveals the proximity between metrics from eye-tracking data. (b) shows the order in which metrics are stacked in the similarity matrix following a second-order Hilbert curve. (c) shows 16 colors with varying hue selected to represent each similarity value in the sub-grid. (d) shows how a Hilbert curve is capable of preserving locality in colors. (e) shows a sample of multi-dimensional stacked metrics with dimension 8×8 for eight participants.

By rearranging the axes (metrics), it is possible to swiftly find those that exhibit similar or identical behavior indicated by parallel line segments between neighboring axes. This is the case for *FixRate*/*SacRat*, *AvgSacDur*/*AvgSac*, and also *FixNum*/*SacNum* (annotations ⓑ in Figure 4.13), enabling an analyst to discard metrics that convey the same kind of information. Furthermore, it is possible to find metrics that exhibit an inverse behavior, which is the case for, e.g., *SacRat*/*AvgFix*, *FixRate*/*AvgFix*, *AvgFix*/*AvgSacDur*, and *AvgSac*/*FixNum*. An increase of the first attribute of each tuple leads to a decrease of the second attribute. These findings reflect the well-known relationships between metrics, which are described in Section 2.2.2.

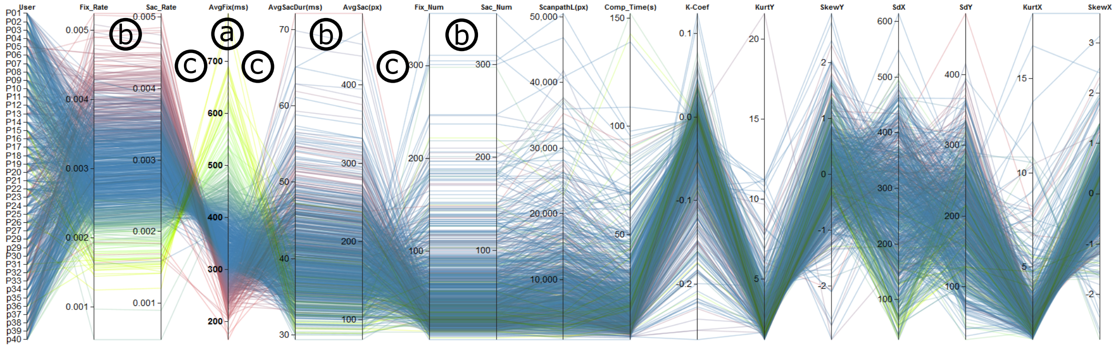
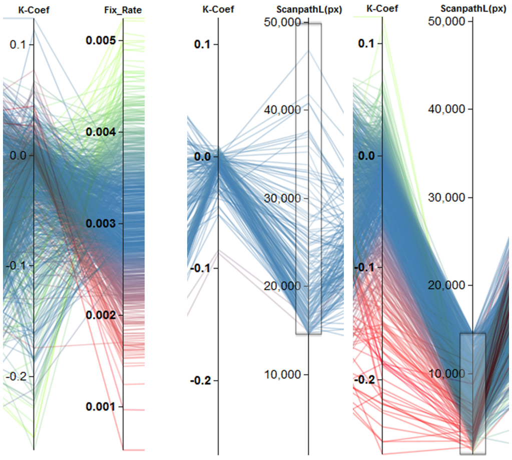


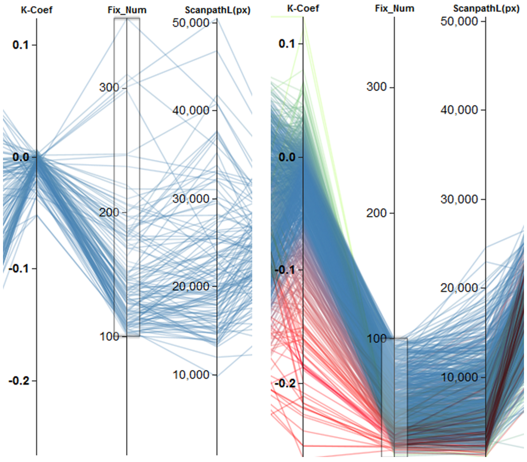
Figure 4.13 — Overview of all variables in a parallel-coordinates diagram. Color indicates different clusters obtained from a clustering based on the fourth axis (*AvgFix*, ⓐ). Axes reveal positive correlation between attributes (ⓑ) and also negative correlations (ⓒ).



(a) (b)

Figure 4.14 — Examples of finding effects based on the clustering of a single attribute. (a) shows the negative correlation between *KCoef* and *FixRate*. (b) depicts two different groups of behavior regarding the *KCoef* based on *scanpath length*. Scanpaths longer than about 15.000 px ((b), left) lead to *KCoef* values centered around zero. Shorter scanpaths ((b), right) result in a broader negative distribution.

Another way to find similarities between attributes is by selecting a specific axis that will cluster the data based on the selected metric, and a color will be assigned to each data instance. Considering the color of the lines as well as the order of the color at other axes, similar behavior can be seen. This is shown in Figure 4.13, where clustering was performed based on *AvgFix* (see annotation ⓐ), resulting in the same color gradient on the axes *FixRate* and *SacRate*.



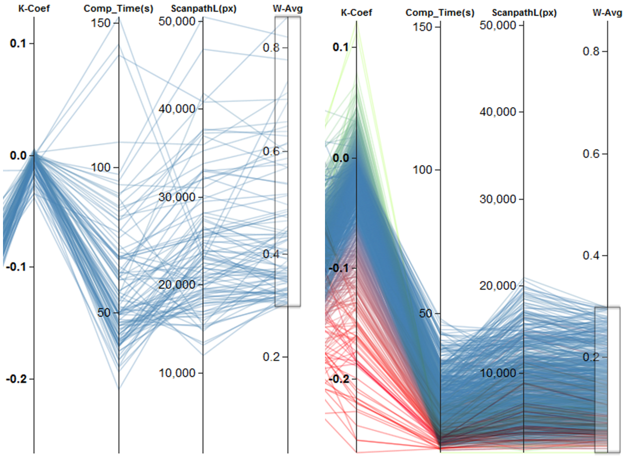
(a) (b)

Figure 4.15 — Separation of participants into two different groups of behavior regarding the *KCoef* based on the number of fixations. (a) shows all subjects with more than 100 fixations. They lead to *KCoef* values centered around zero. Fewer fixations result in a broader negative distribution, which is shown in (b). This achieves about the same separation as utilizing the *scanpath length* with a threshold of about 15.000 px.

Using clustering based on a single metric, an inverse relationship between *KCoef* and *FixRate* can be found (see Figure 4.14a). This indicates that the ambient/focal attention changes inversely to the fixation rate. Furthermore, ambient/focal attention varies with the *scanpath length*. As seen in Figure 4.14 for a length of more than about 15.000 px, the distribution of the $\mathscr{K}$ coefficients is centered around zero (ambiguous cases), whereas for shorter scanpaths the distribution is more widespread and tends toward negative values (ambient attention).

The $\mathscr{K}$ coefficient is also related to the number of fixations in a similar fashion. Here, the same kind of distributions can be observed for more than 100 fixations (ambiguous cases) and fewer fixations (ambient attention). If the scanpath length is considered, it can be seen that the separation into two groups having more and less than 100 fixations yields roughly the same separation as splitting the subjects according to a scanpath length of more or less than 15,000 px. This is shown in Figure 4.15.

Another interesting question is how the $\mathscr{K}$ coefficient is related to the *completion time* and *scanpath length*, since there could be, e.g., four subgroups of participants using a categorization of fast/slow for the *completion time* and short/long for the *scanpath length*. Previously, the *scanpath length* and the $\mathscr{K}$ coefficient were



(a) (b)

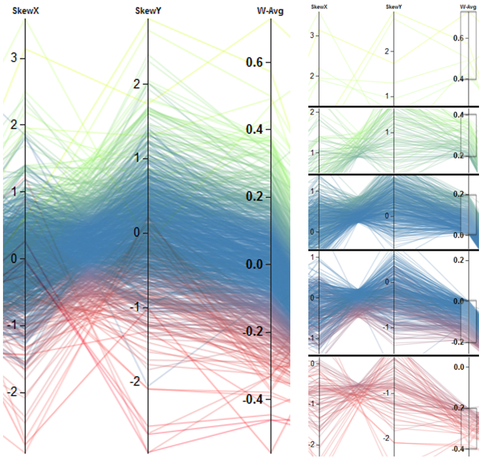
Figure 4.16 — Combining completion time and scanpath length to generate a new 1D metric (*W-Avg*). (a) shows that values above a threshold of 0.3 of *W-Avg* result in one group of participants that exhibit ambiguous gaze behavior, whereas (b) depicts the second group that is looking all over the stimuli (ambient gaze behavior), according to the $\mathscr{K}$ coefficient.

analyzed. Therefore, the focus can now be switched to the influence of the *completion time* assigning a weight of 0.7, while a weight of 0.3 is assigned to the *scanpath length*. The results are shown in Figure 4.16. Based on this affine combination of metrics, similar results are achieved compared to utilizing the *scanpath length*: One group of participants exhibits ambiguous eye movements, whereas the other groups show more ambient eye movements.

Besides the so-far used eye-tracking metrics that characterize user behavior, identifying spatial behavior is also of interest. Therefore, metrics can be analyzed that give an impression about regions participants were looking at. The skewness of the distribution of positions, projected onto the x- and y-axis, can be used for getting a rough impression. Figure 4.17 shows the combination of *SkewX* and *SkewY* with an equal weight of 0.5 for each metric.

Overall, there is a negative correlation between the metrics, which is also the case if different intervals of values of the combined metric are considered (see Figure 4.17b).

To further investigate the spatial distribution, kurtosis can be included in the analysis. Based on the skewness, approximate areas are derived where participants were looking at. The kurtosis of the distributions describes now whether there are frequent small deviations of mean values or frequent larger dispersed data. The first case would indicate the existence of focus attention

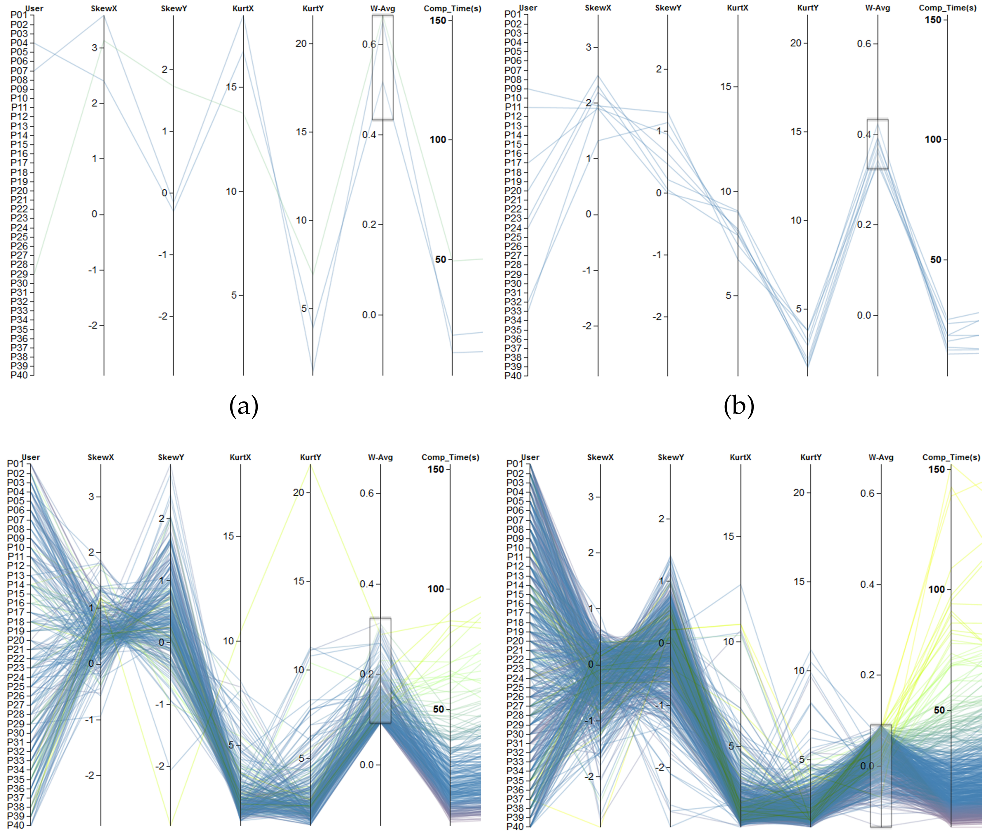


(a) (b)

Figure 4.17 — Color-encoded results are shown in (a). Clustering was performed on the weighted average of the first two columns. Overall, a negative correlation can be recognized. Selecting different values ranges from the weighted average lead to the selection of subgroups of similar skewness values. The weights were set to 0.5 for the two metrics. All subgroups also exhibit a negative correlation. This is shown in (b).

areas, whereas the later a wider spread of attention on possibly multiple locations. Since the effects of kurtosis are of more interest, these metrics got a higher weight (0.7), and skewness got a lower weight (0.3). Based on the newly combined metric, participants can roughly be separated into four groups. Each group shows distinct visual patterns. The first one (Figure 4.18a) contains only a minority of the data that shows large changes between neighboring axes. The second group (Figure 4.18b) exhibits a drop-off of the values on some axes from left to right. The third (Figure 4.18c) and fourth (Figure 4.18d) group show similar values for the kurtosis in both spatial directions and also for the skewness in x-direction, but differ visually in the skewness in y-direction. With this kind of analysis, using parallel coordinates, characteristics of each scanpath can be investigated for each participant during the *Metric Analysis* step of the workflow (see Figure 4.9) and preliminary clusters can be formed based on the color encoding of the lines and similar visible line structures.

During the following *Participant Group Analysis*, the multidimensional stacking approach allows one to compare all participants to each other on a more abstract level. Standard approaches generate a similarity matrix where the columns and rows represent the participants, who are ordered alphanumerically according to ID. Each cell of the matrix contains a similarity value. Here, a matrix cell

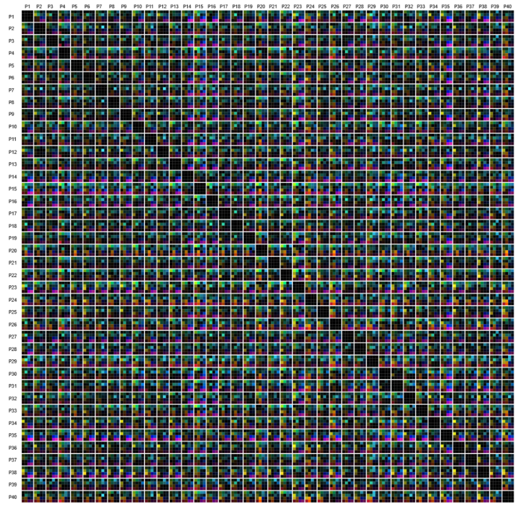


(a) (b)

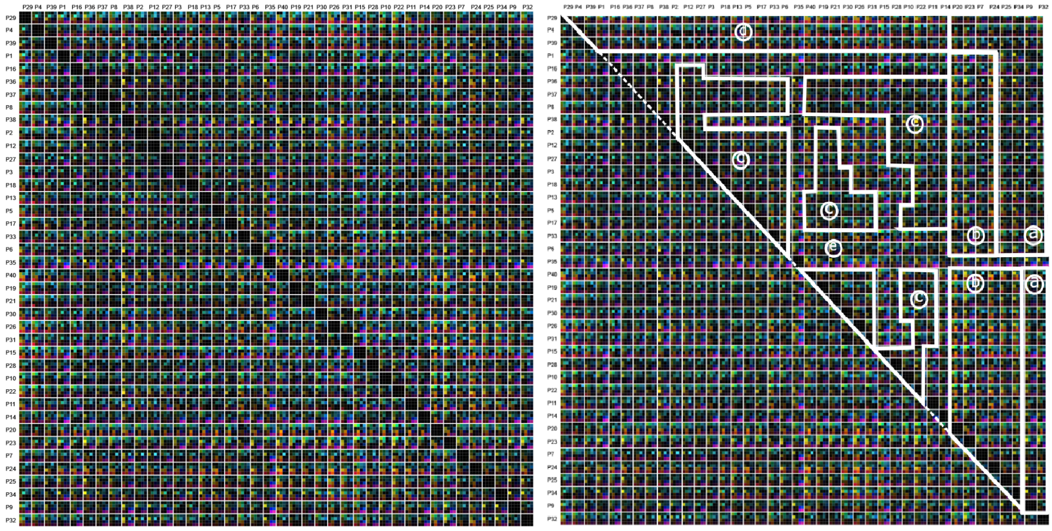
(c) (d)

Figure 4.18 — Combination of four metrics (*SkewX*, *SkewY*, *KurtosisX*, and *KurtosisY*). Color encoding is based on the clustering according to *completion time*. The four images show a rough categorization of the participants into four groups. Metrics values in (a) show large changes between neighboring axes, whereas in (b) there is a visible drop-off of values for some axes. Images (c) and (d) indicate same values with respect to kurtosis, but differ in *SkewY*. The weights for skewness were set to 0.3, whereas a weight of 0.7 was assigned to the kurtosis-related metrics.

contains several color-encoded elements that correspond to the similarity values of the used metrics. An example of this is shown in Figure 4.19a. Here, isolated group structures can be seen, but in general, the visualization looks rather chaotic. By performing a reordering of the participants, an order is introduced to the chaos, to some degree: getting similar participants spatially closer reveals more group structures than before. Figure 4.19b depicts the similarity matrix with reordered participants, whereas Figure 4.19b highlights a possible grouping. The new order of participants is the result of hierarchical clustering based on a combined metric (skewness in x- and y- direction with



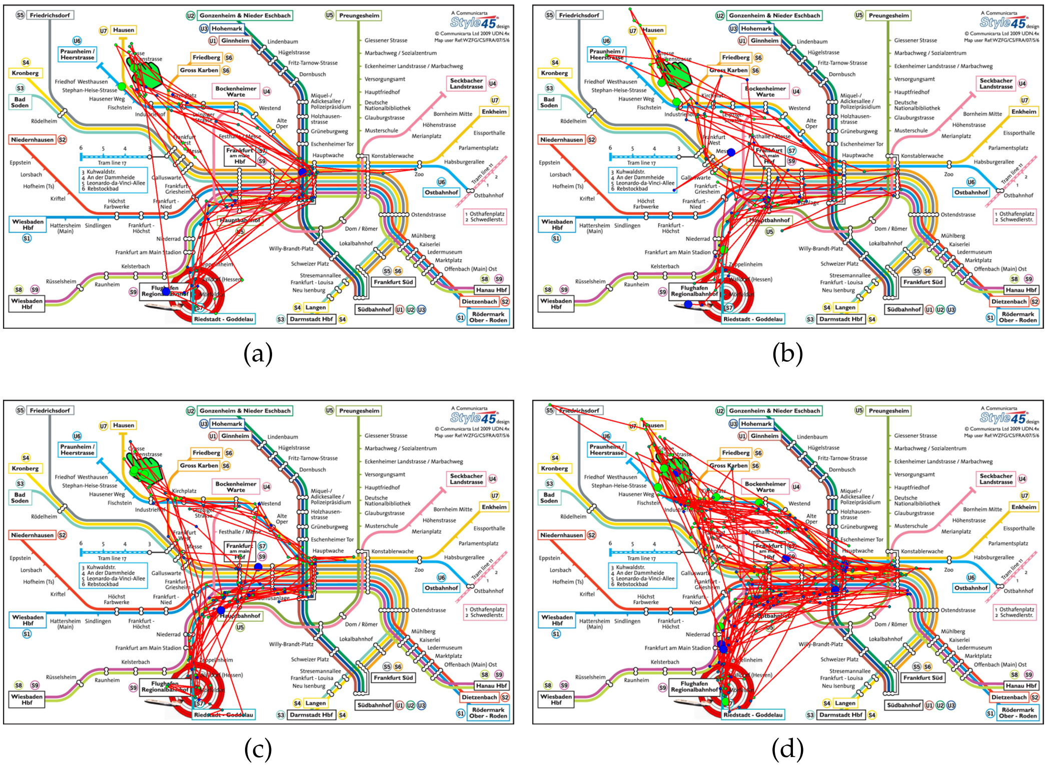
(a)



(b)

Figure 4.19 — Similarity matrices including all participants. All 16 similarity metrics are depicted by color. (a) shows the original order of the participants, which is increasing from left to right and top to bottom, according to the participant ID. For (b), the order of the participants was changed according to the result of the clustering based on the newly combined metric (skewness in x- and y- direction with a weight of 0.3; kurtosis in x- and y- direction with a weight of 0.7). By visual inspection, six clusters emerge (ⓐ–ⓔ), highlighted by white lines.

a weight of 0.3; kurtosis in x- and y- direction with a weight of 0.7). By using the new ordering, six different groups are identified exhibiting same visual characteristics.


(a) (b)

(c) (d)

Figure 4.20 — Examples of scanpaths for participants that were confronted with color-encoded metro maps. (a) shows scanpaths of participants who are contained in group B. (b) shows scanpaths of participants who are contained in group A. (c) and (d) show scanpaths of participants from group A and B. Scanpaths in (c) are more similar to (a), since they are focused on a path leading from the start to the main station and then to the destination. Scanpaths in (d) are more similar to those in (b), since they are less focused on one specific path, and involve areas left and right of the main station. The start location is designated with a green hand and the destination with a target symbol. The start of the scanpath is indicated by a big green dot and the end of the scanpath by a blue dot. The color of smaller fixations in between is gradually changing from green to blue, encoding the temporal ordering of the fixations.

Inspecting the different groups in more detail using scanpaths as part of a qualitative evaluation, it can be seen that, e.g., group A and B differ in defining characteristics of scanpaths that were investigated by the participants. Selecting scanpaths of participants within group B (see Figure 4.20a) shows that there is a designated path that was investigated, whereas scanpaths of participants in group A (see Figure 4.20b) are more diverse. Comparing these scanpaths with those of participants that are part of both groups (see Figure 4.20c and (d)), it can be seen that the scanpaths share similar characteristics, thus the scanpaths are separated into two groups exhibiting two defining characteristics: a more focused versus a more diverse or widespread investigation.

Part II

Empirical Evaluation Using Eye Tracking

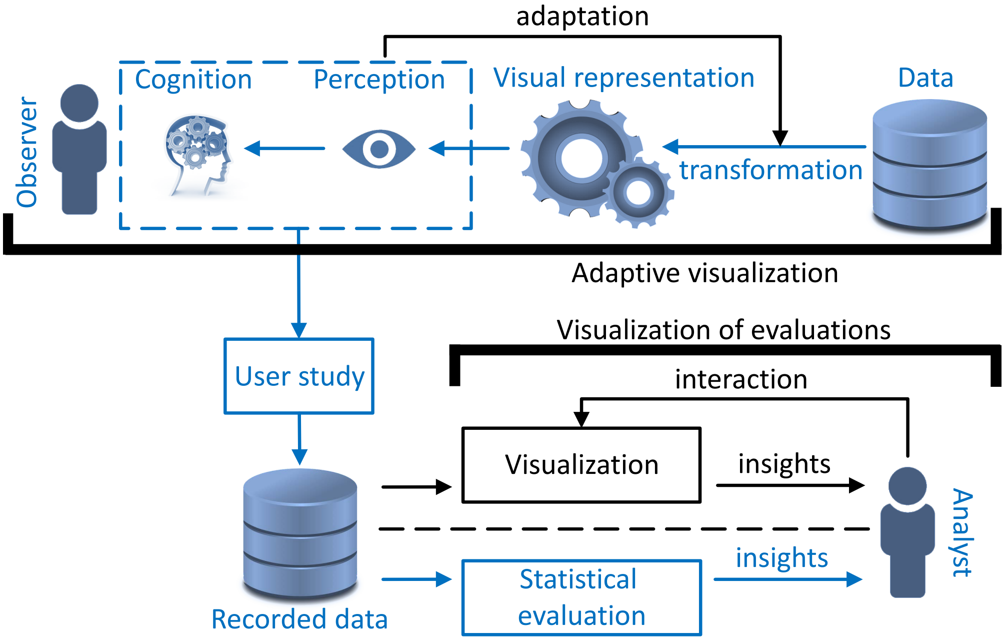


Figure II — Components and workflows involved in Part II.

In contrast to the first part of this thesis, where approaches for visualization for data analysis and also adaptive visualizations were presented, this second part addresses the empirical evaluation by means of eye-tracking studies. Figure II exhibits the components (highlighted in blue) of the introduced overarching workflow that are relevant in this second part of this thesis. Similar to the first part, representations (stimuli) are generated, but they are not automatically analyzed using models. Instead, they are presented to observers who solve a given task utilizing their perception of the stimulus. Eye-gaze and other necessary data are recorded, and they are later used in a statistical evaluation that is the most important highlighted component of the workflow. Depending on the stimulus and task, specific evaluation approaches are utilized to generate insights into behavioral patterns or used solution strategies.

The domains covered by the user studies are as followed: First, trajectories and their direction encoding were tested, since lately trajectories are frequently used in visualization. One probable cause for this is the fast technological progress that allows scientist nowadays to use small and light-weight position tracking devices such as GPS to record time-dependent position information of different kinds of objects or living beings. Direction encoding has already been researched for node-link diagrams [123, 124], but not in the context of trajectories. Here, it is not easy to follow aesthetic criteria for graph drawing [125, 126], since node-link visualizations of trajectories have very different characteristics regarding the distribution of node positions and link lengths: both are less evenly distributed than for abstract graph drawings. Furthermore, trajectories have only one incoming and one outgoing link at the most but no arbitrary valence. Additionally, different tasks are relevant for trajectories compared to node-link diagrams, and with given restrictions of geographic positions, they tend to suffer from visual clutter and overdraw in dense or complex regions.

Second, the topic of correspondence visualization and visual search using annotations. These are required in many tasks. Especially in the context of visualization, where it is not always possible to provide appropriate interaction methods that support analysts in their tasks. To perform a visual search, a reader has to actively scan the visual environment, e.g., a map filled with textual descriptions. All elements that are not of interest are so-called distractors. The search for a certain feature becomes more difficult with increasing number of distractors, unless the search target *pops out*. This phenomenon is researched in preattentive processing [127, 128]. In contrast to previous work, a visual search that is not supported by preattentive processing is investigated here. Related to this issue is the typical problem of searching for a target within a group of perceptually similar distractors, which is being considered here with a focus on perceptual processing. Additionally, a model was derived that reflects the utilization process of different archetype annotation methods.

Third, it was analyzed how people utilize maps, or more specifically metro maps, which are not easy to design [11]. These kinds of maps were used since people can relate to such kind of stimuli and results of the study could lead to improved metro maps, which might be helpful to, e.g., tourists and others that are new to public transport. It is necessary to extract reading behaviors and incorporate them into the design process to achieve this.

Finally, based on eye tracking, behavioral patterns of participants were inferred while solving a generic perceptual task using basic techniques for the visualization of multi-dimensional data. The evaluation is related to a preliminary study [129], which used a densely sampled parameter space and a distance estimation task that is fundamental to many other high-level tasks [130].

CHAPTER 5

Trajectory Encoding

Trajectories are commonly visualized by using line-based representation: connecting sample points along trajectories. Without further information, a typical interpolation model relies on a linear connection between sample points, yielding a piecewise linear representation of the trajectory. Such visualizations are, e.g., used in animal movement ecology [131, 132] to get an overview of movement behavior, or spot clusters that could indicate nesting sites or hunting grounds.

Performing low-level tasks like following the path of a single trajectory comes with low perceptual and cognitive effort for the user as long as short and simple trajectories are shown. However, performing path-oriented tasks for complex and dense data can become hard and error-prone. Therefore, there is potential for improved visualization techniques to increase the user performance for such complex data sets.

The goal of this chapter[1] is to evaluate and compare different trajectory visualization techniques. All of them follow the same fundamental principle of directed node-link diagrams: nodes represent sample points along the trajectories, directed links connect subsequent sample points. Node-link diagrams were used because they follow the congruence principle for effective graphics [90]:

---

[1] **Parts of this chapter have been published in:**
R. Netzel, M. Burch, and D. Weiskopf. Comparative eye tracking study on node-link visualizations of trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2221–2230, 2014 [6], © 2014 IEEE.

the structure and contents of the diagram fit to the structure and content of the internal and mental representation that the visualization tries to achieve. Furthermore, other than more abstract visualizations, they lend themselves for direct overlay on geographic maps that serve as an important spatial context for trajectories.

As concrete choices of visualization techniques, the following methods for showing directed links are being evaluated: standard arrow, tapered, equidistant arrows, and equidistant comets. The first two are standard approaches known from node-link diagrams for directed graphs; the latter two are variants of existing techniques that were developed to show links of largely varying length in a single diagram. The first technique simply puts a standard arrow at the end of a link, as most commonly used in node-link diagrams. Tapered links performed previously very well in node-link diagrams of abstract directed graphs [124], and the human ability to perceive direction inspired their design. Derived from the standard arrow, the equidistant-arrows technique consists of multiple arrows along a link that are placed equidistantly. Finally, the equidistant-comets technique places a *comet texture* at equidistant positions along a link to show direction information along the full length of the link. These link types are depicted in Figure 5.1. In addition to the link rendering style, the effect of rendering order is being investigated for the halo visualization of the links; halos are used to visually separate crossing links, which often appear in dense and complex data sets. Finally, the effectiveness of node splatting for large data sets is tested as an alternative approach to visualize point locations that provide visual feedback about the point density. All combinations of link visualization techniques are tested for two different levels of trajectory density. The comparison is based on a repeated-measures within-subjects user study with 25 participants, conducted in a controlled laboratory environment. Task performance was tested as a measure of visualization effectiveness. As tasks, three representative types were chosen: tracing of paths, identification of longest links, and estimation of the density of trajectory clusters. Task performance was measured by task completion time and accuracy. Also, the study was set up as an eye-tracking experiment to understand the reasons for possible differences between visualization techniques better. The eye-tracking data was analyzed by using average fixation duration and saccade length as metrics.

The main result of the study is that tapered links perform well, even under different data set conditions and tasks. However, depending on the zoom level and variability of link length, the equidistant comets and arrows can show some benefits, too. Additionally, the use of point splatting turned out to be beneficial.

5.1 Related Work

In many situations, it is necessary to encode directional information, and there is a variety of options that can be selected. The question that arises is which option should be chosen. Different studies were conducted to answer this question. Holten et al. [124, 123] performed two studies concerning the direction encoding of links of node-link diagrams of directed graphs. In their first study, they tested six different link types: standard arrow, color gradient from green to red and light to dark and vice versa, curved links, and tapered links. Here, tapered links outperformed all other types concerning solution time as well as accuracy. In their extended study, they tested 15 link types including the previous ones. Additional types were glyph-based, or they used animation. Again, tapered links outperformed all non-animated types in solution time as well as accuracy, whereas animation further increased the accuracy, at the cost of a slightly increased completion time.

Concerning node-link diagrams of abstract graphs, other papers address the readability of direction encodings in the presence of link crossings. Burch et al. [133] investigated in their study the readability of directed graphs shown with partially drawn links instead of links of full length. This approach reduces link intersections and can lead to shorter task completion times but at the cost of higher error rates. Bruckdorfer et al. [134] investigated the influence of different parameters in visualizations with partially drawn links. Jianu et al. [135] proposed color encoding of each link according to a closeness metric to address the problem of link crossings. Rusu et al. [136] introduced breaks in links if intersections occur, leading to an effect similar to partially drawn links. Since such partially drawn links showed high error rates and were designed for the visualization of abstract graphs, they were not considered as visualization technique in this study. Instead, a node-link visualization was chosen because it is very common for representing spatial trajectories in fields like geographic information science [137] and movement ecology [131, 132].

The most common encoding of direction information uses arrows. Even with trajectory clustering and bundling, arrow-style rendering is often employed [138]. Recently, Janetzko et al. [139] have described a method to simplify trajectories and used tapered links for direction indication. Therefore, arrows and tapered links were selected as the two representatives of traditional rendering techniques for trajectories, to be tested in this user study.

Especially for dense and large data sets of trajectories, some approaches create density fields of trajectories. By convolution of trajectories with a low-pass filter, a visualization can be achieved that highlights frequently visited areas [140]. Density maps can also be used to explore multivariate data of trajectories [141].

Similarly, edge splatting can be employed to construct a density field of links in a node-link diagram [142]. Splatting or a density representation of links was not used because they tend to obscure direction information in order to highlight high position density. However, splatting of nodes was included—similar to graph splatting [143]—to show the positions of samples along trajectories.

The encoding of direction information is not only important for node-link diagrams and trajectories, but also in other domains like flow visualization. Flow visualization often encodes direction by glyphs. Pilar et al. [144] investigated this approach in detail. Other techniques use texture-based methods [145] to visualize flow direction. The comet texture is inspired by oriented line integral convolution (OLIC) [146] and follows the recommendation for stroke design in flow visualization [147]. However, it should be pointed out that flow and trajectory visualization exhibit many important differences: streamlines, pathlines, etc. in flow visualization are typically smooth and curved, not piecewise linear; streamlines do not intersect, and pathlines do not have the intersection characteristics of trajectories; finally, sample positions along streamlines and pathlines are not visualized.

In general, there are only a few previous eye-tracking studies in information visualization. The following examples use eye tracking to test techniques that do not deal with trajectories or graphs: Goldberg et al. [148] described general aspects of eye tracking and performed a comparison between linear and radial plots. Burch et al. [149] investigated differences of traditional, orthogonal, and radial node-link tree layouts. Garlandini et al. [150] tested different visual variables for the visualization of geographic maps and performed an empiric evaluation of recorded eye-tracking data.

5.2 Visualization Techniques

In this section, variants of visualization techniques are described that were compared in the user study. First, four types of rendering directed links are discussed, followed by an explanation of halo rendering with depth sorting. Finally, different ways of visualizing nodes are described.

Throughout this chapter, abbreviations are used for the different options of link rendering, sorting, and node visualization. These abbreviations are summarized in Table 5.1 and Table 5.2.

Table 5.1 — Abbreviations for the visualization methods.

N	No direction encoding
A	Direction encoding with standard arrow
EA	Direction encoding with equidistant arrows
C	Direction encoding with equidistant comets
T	Direction encoding with tapered links
RND	Random edge sorting
SF	Shorter links on top of longer links
LF	Longer links on top of shorter links
SP	Splats for point visualization of nodes
NSP	No splats for point visualization of nodes

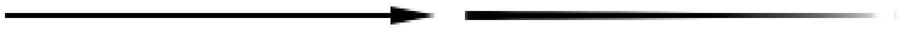
Table 5.2 — Abbreviations for different sizes of data sets used in the study.

D1	Low data density (3 trajectories with 30 points each)
D2	High data density (3 trajectories with 60 points each)

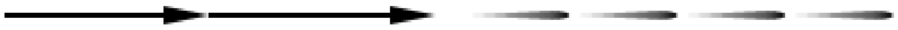
5.2.1 Link Types

The simplest and most common type of link visualization uses the standard arrow (A). Here, the end arrow is rendered to encode direction information, see Figure 5.1a. Another representation is based on tapered links (T): the direction is encoded by a prolonged arrow between the start position (thick end) and the end position (arrow tip), see Figure 5.1b. Standard arrows were included because of their widespread use, and tapered links due to their very good performance in previous studies on node-link diagrams of abstract graphs [124, 123].

The node-link visualization of spatial trajectories comes with additional challenges compared to the visualization of abstract graphs. In particular, there can be large variations in link length. Furthermore, data sets can become large and complex, requiring zooming for a detailed visual analysis. Unfortunately, standard arrows and tapered links are not well suited for very long links. As illustrated in Figure 5.2, direction information might be completely missing for standard arrows, if neither their start nor end node is shown (due to zooming). Similarly, the thickness gradient for tapered links might become very small, making it hard to perceive direction information. To address this problem, two new variants of link representation were developed. The first technique is called equidistant arrows (EA), which is an extension of the standard arrow: it places multiple arrows along a link at equidistant positions, see Figure 5.1c. The distance between arrows is adapted to the zoom level and identical for all

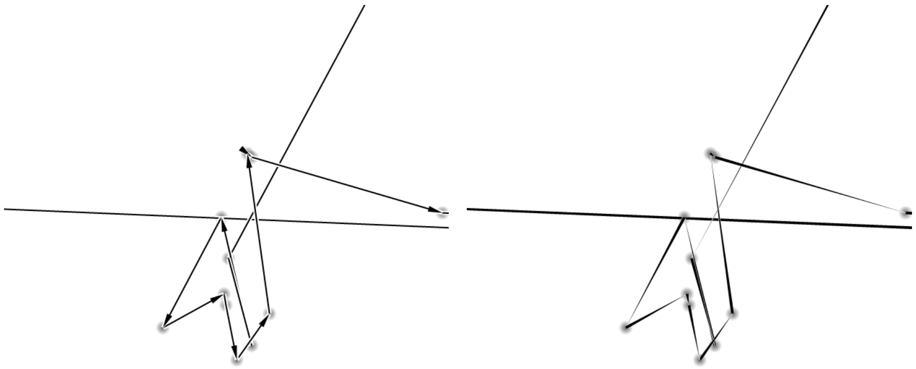


(a) standard arrows (b) tapered link

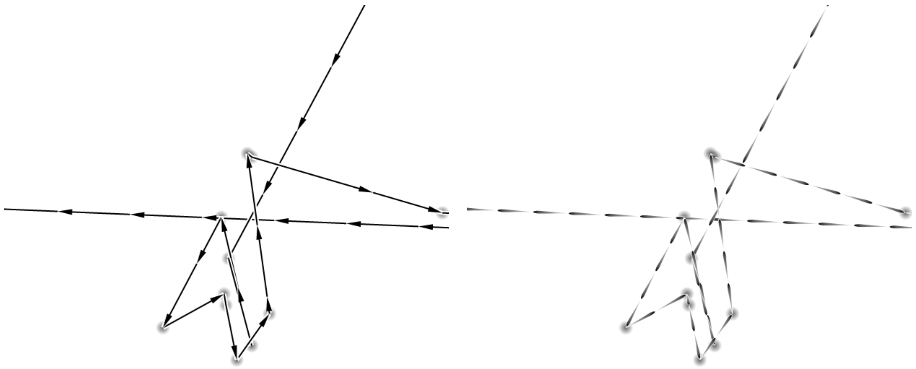


(c) equidistant arrows (d) equidistant comets

Figure 5.1 — Types of link visualization.



(a) standard arrows (b) tapered links



(c) equidistant arrows (d) equidistant comets

Figure 5.2 — Readability of link direction with different types of link visualization, for a zoomed-in view.

links of a zoom level—except for the quantization effect introduced by placing an integer number of intervals along each link.

The second developed type is the equidistant comets (C) that is suitable for very long links, see Figure 5.1d. The strategy is the same as for the equidistant arrows. By using a repeating direction indicator along a link, it is easy to perceive the direction at any position or scale. The metaphor of a comet inspires the repeated texture: the head of the comet is large, and it is followed by a tail that is becoming narrower. A single comet texture is similar to a scaled-down version of the tapered link. The main difference is that direction is essentially reversed: the direction goes from the narrow tail of the comet stroke toward the large end. This opposite encoding scheme was chosen because it is inspired by stroke-based rendering in flow visualization: Halley [151] used similar strokes as early as in 1686. The visual design with the thick head is also recommended by Bertin [152] and confirmed by a user study by Fowler and Ware [147]. More background of the perception of flow visualization is given by Ware [153]. The comet texture with the opposite direction encoding of tapered links was chosen to extend the design space of this user study compared to Holten et al. and because the equidistant placement along links resembles the visual signature of flow visualization.

For dense data sets, both arrow rendering methods (A, EA) would suffer from visual clutter and overdraw: important structures like link crossings or other arrows might become occluded. Only those arrows are rendered that do not occlude important structures to address this issue. This is a trade-off because missing arrows also imply the absence of direction information.

5.2.2 Depth Sorting

Halo rendering of links was employed to improve the perception of them at crossings. Therefore, the rendering order plays an important role: links in the background are *cut-through* by halos of foreground links. To this end, the effectiveness of different depth sorting methods was studied by comparing random render ordering (RND) to ordered rendering. The order, respectively the depth, reflects the importance of links. More important links will be rendered over less important ones. In this study, the total link length between nodes is used as importance measure. Here, an ordering that puts the shortest links in the foreground (SF) and one that places the longest links in the foreground (LF) was evaluated. Figure 5.3 illustrates the effect of depth sorting for halo rendering.

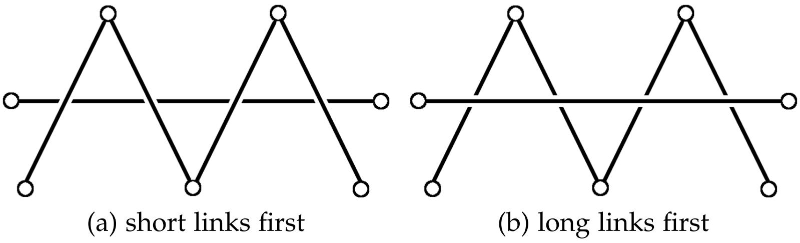

(a) short links first (b) long links first

Figure 5.3 — Effects of the ordering in combination with a halo rendering.

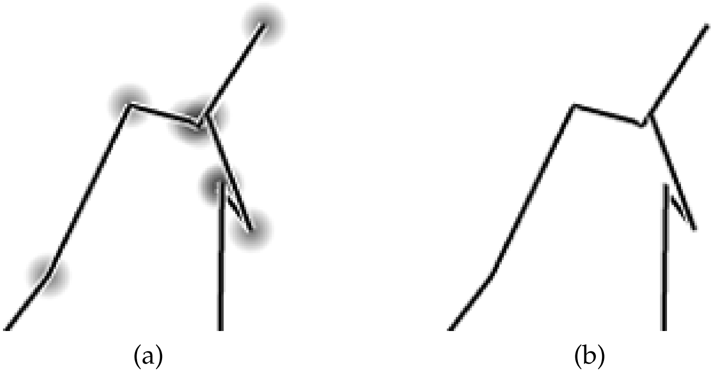

(a) (b)

Figure 5.4 — (a) Node visualization by splatting, (b) implicit node representation (indirectly through the end points of links).

5.2.3 Node Visualization

The nodes of a node-link diagram are shown by some kind of point visualization. Typically, these points are rendered as non-filled circles, filled dots, or they are not explicitly shown (only implicitly as ends of links)—the latter being abbreviated as NSP. However, for large and dense data sets, opaque rendering of circles or dots would lead to much overdraw and clutter. Therefore, point splatting (SP) was included, similar to the splatting of vertices in graph splatting [143]. Here, each node is rendered as a circular splat with a radially decreasing intensity. Using additive blending to render each splat, results in a density field. Regions with many nodes that are very close to each other will lead to high values in the density field and thus can be identified easily. In contrast, using dots to indicate nodes would lead to much overdraw; in some cases, a user might perceive a single dot although this dot might occlude dozens of other node representations. Figure 5.4 illustrates SP and NSP.

5.3 Experiment

This section describes the design, hypotheses, tasks, stimuli, and further details of this user study.

5.3.1 Design

The user study follows a repeated-measures design. The study was conducted in a controlled laboratory environment; eye-tracking data was recorded during the study. The study consisted of three independent parts. The first two parts were designed to compare the different rendering methods for links. Those two parts had the following within-subjects factors to represent the different visualization techniques: link type (A, EA, C, T) and sorting order (RND, SF, LF). Also, two different data set density (D1, D2) were tested for each combination of techniques (see Section 5.3.4). The third part of the study was designed to evaluate the effect of node splatting. It had the following within-subjects factors: splatting or no splatting for point rendering (SP, NSP).

The three parts used the same dependent quantities that served as a basis to assess the effectiveness: task solution accuracy, task completion time, average fixation duration, and average Euclidean saccade distance (the latter two were acquired through eye tracking).

5.3.2 Hypotheses

The hypotheses are based on previous results from user studies on node-link diagrams of abstract graphs [123, 124] and prior experience obtained from working with trajectory visualization for movement ecology:

H1: Tapered links (T) outperform the other link representations for tasks related to following trajectory paths.

H2: Equidistant comets (C) outperform the other link representations for tasks related to recognizing very long links.

H3: Depth sorting (SF) improves correctness of following trajectory paths in dense regions because short links are rendered on top of longer ones.

H4: Depth sorting (LF) improves correctness of tasks related to recognizing very long links because long links are rendered with guaranteed visibility.

H5: Point splatting (SP) improves the perception of high-density clusters of nodes because it provides a more accurate representation of node distribution than the visualization by the links alone.

Hypotheses H1–H4 refer to the first two parts of the study (related to link rendering), hypothesis H5 refers to the third part of the study (related to node splatting).

5.3.3 Tasks

Two tasks (Task 1 and Task 2) were designed to compare the techniques for link visualization and one task (Task 3) to evaluate the effect of node splatting. Those tasks were chosen based on similar tasks used in the visualization of abstract graphs [154, 155] and based on tasks specific to the analysis of movement patterns [156]. While the tasks of this user study are presented as abstract tasks to the participants, they have concrete and realistic correspondences to examples in movement ecology: the tasks could be used to identify behavioral patterns like migration and foraging of animals. Nevertheless, the tasks are generic in the sense that they carry over to many other application domains.

Task 1 (Path Following): The first task was to follow the path of a trajectory from a given start point. In particular, participants were asked to answer the following question for the shown stimulus: *Is there a way from start to end in only five hops?* The start and end position were color-coded (start=blue, end=orange) in a way they were easy to recognize. The dots for the two positions were shown alone—without the trajectory visualization—for two seconds before the full visualization was presented. In this way, no time for visual search of the two nodes was required, restricting the task to following the path of the trajectory. The participants were instructed to answer as correctly and fast as possible; they were told that there was no time limit (and thus no pressure of time). The participants provided their answer by pressing a respective key. The task completion time was measured as the temporal difference between pressing the key and the first appearance of the full visualization. Figure 5.5 shows the screenshot of an example of Task 1.

Task 2 (Longest Link): The second task was to identify the longest link. This task was chosen because it tests whether the full visualization can be read effectively, making it complementary to the path tracing of Task 1. In particular, participants were asked: *Select the longest link by clicking on it.* Again, the participants were instructed to provide correct and fast answers, and that there was no time limit. Task completion time was measured as the temporal difference between mouse click and the first appearance of the visualization. Accuracy was measured by checking whether the mouse click was on the correct link, or not. The longest link was surrounded by a virtual rectangular bounding box to avoid unintended errors from slightly wrong mouse positions. The width of the box was twice the link width. A click within the bounding box

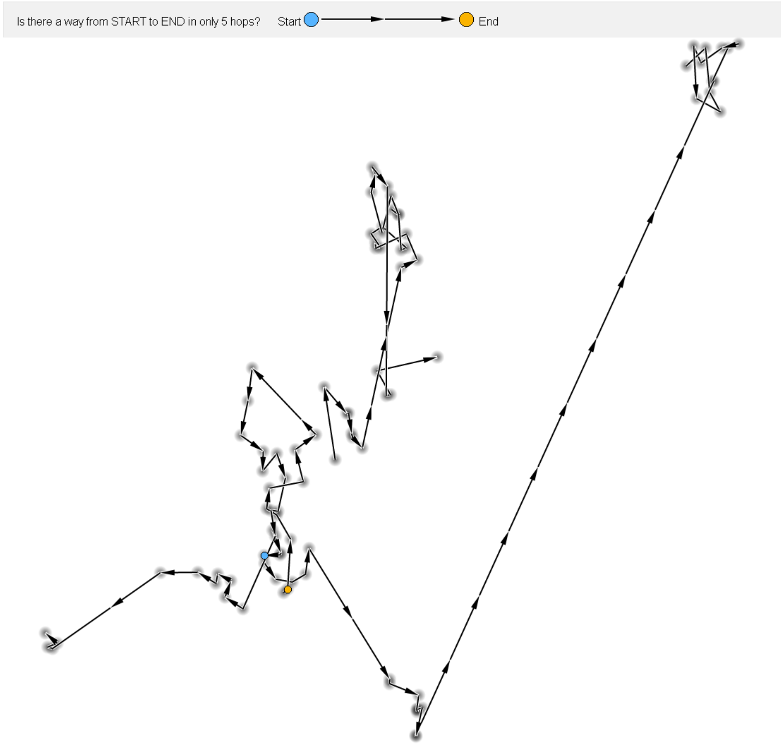
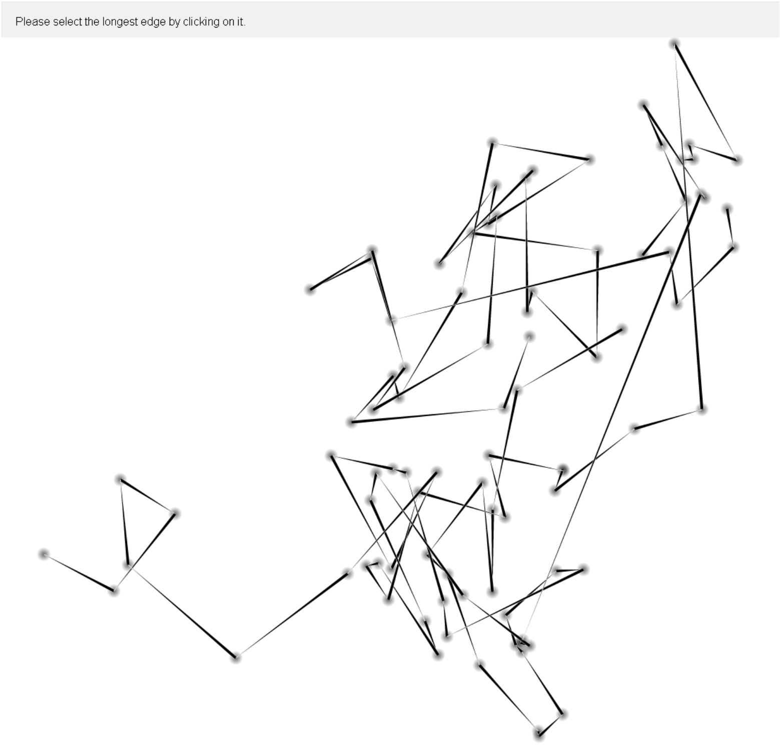


Figure 5.5 — Screenshot of an example of Task 1 (path following). Here, equidistant arrows (EA) with node splatting (SP) and no depth sorting (RND) are shown. The legend at the top contains the question and the description of start and end nodes.

was recognized as a correct answer. Participants were instructed to not click on intersections of links to avoid ambiguity in the identification of the clicked link. Figure 5.6 shows the screenshot of an example of Task 2.

Task 3 (Number of Nodes in Clusters): The third task was designed to test the visualization techniques for nodes. Here, it was of interest how well participants could identify the overall structure of the spatial distribution of nodes. To test this, participants were asked: *Select the cluster with the most elements by clicking.*

Subjects were encouraged to make their decision according to the visual appearance of the clusters, not by explicitly counting nodes, because it was of interest how well the participants could obtain a rough impression of the distribution of nodes. Again, the participants were instructed to provide correct and fast answers, and that there was no time limit. The task completion time was measured as the temporal difference between mouse click and the first appearance of the visualization. Accuracy was measured by checking whether the mouse click



Please select the longest edge by clicking on it.

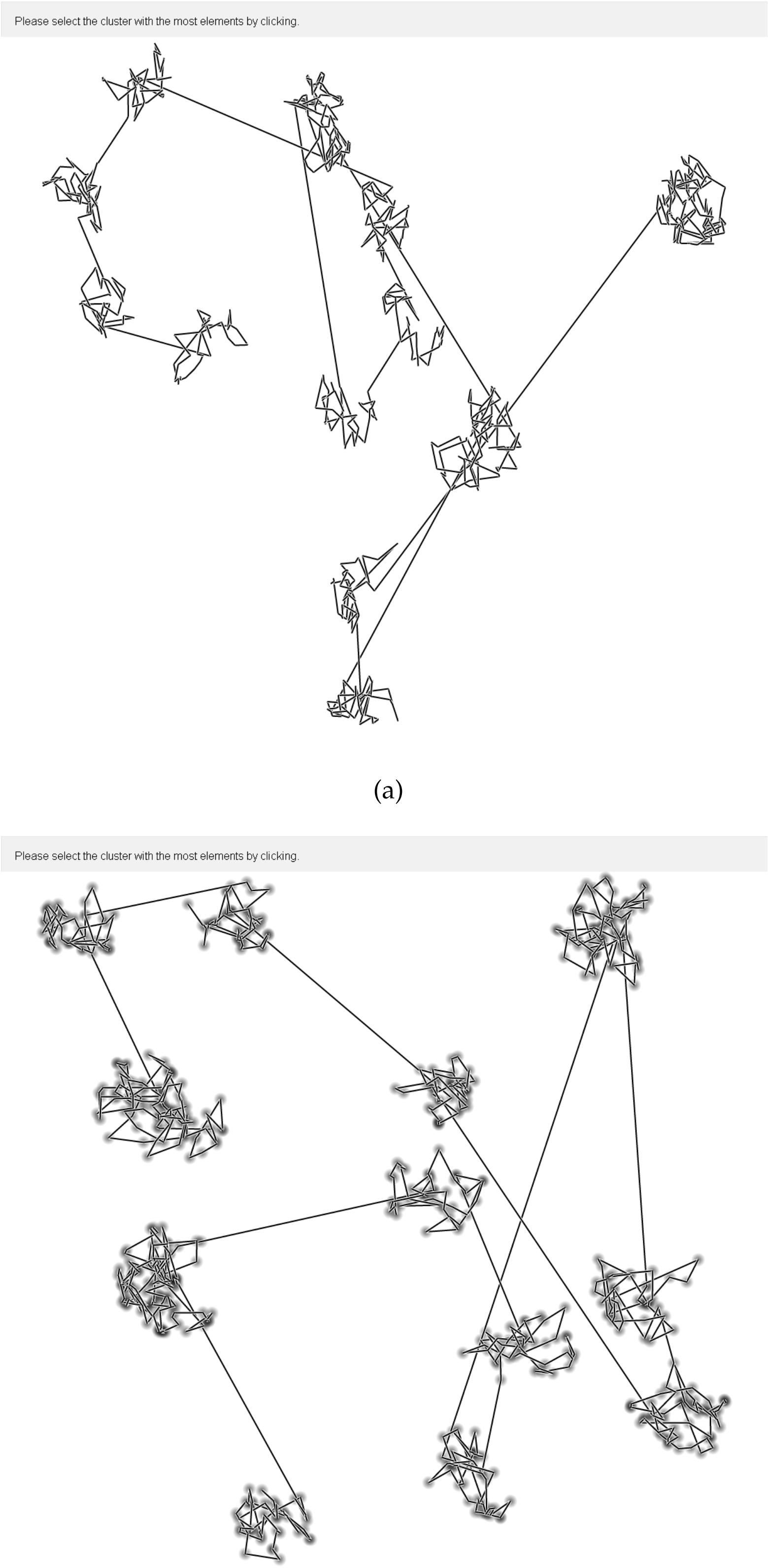
Figure 5.6 — Screenshot of an example of Task 2 (longest link). Tapered links (T) with node splatting (SP) and depth sorting (LF) were used. The legend at the top contains the question.

was in the area of the correct cluster. Figure 5.7 shows two example screenshots for Task 3.

5.3.4 Stimuli

The visual stimuli were images rendered with the different variants of visualization techniques. The images were generated with 2x anti-aliasing and had a size of 1100 × 1100 pixels. Halo rendering was enabled for all images. The line width was 5 pixels in the virtual high-resolution frame buffer of 2200 × 2200 pixels, i.e., 2.5 pixels in the final image.

The visualization techniques were applied to a generative data model to produce the visual stimuli. A Markov chain model based on real-world movement data from birds was used to generate realistic data: the input data contained the trajectories of three oystercatcher birds with a total number of 16,434 GPS locations. From that data, transition matrices were computed for both the travel distance between GPS locations and the relative change in direction per



Please select the cluster with the most elements by clicking.

(a)

Please select the cluster with the most elements by clicking.

(b)

Figure 5.7 — Screenshots of examples of Task 3 (number of nodes in clusters). (a): no splatting (NSP), (b): node splatting (SP).

location. Each matrix contained a probability distribution for the transition from one state to another, based on the training data. The transition was encoded with 20 states, i.e., each matrix had a size of 20 × 20. The maximum distance between two locations in the data set was around 4,000 m, so each state covered around 200 m. For the change for movement direction, each state covered $360^\circ/20 = 18^\circ$. With this Markov chain model, important characteristics were retained of real trajectories from movement ecology, serving as a basis for a realistic user study.

The stimuli for Task 1 were produced from the generative data model directly. The stimuli were generated so that in 50% of the trials the correct answer for Task 1 was *yes*. A start position was randomly chosen and then an intermediate end position as selected at a random distance of 2 to 5 hops (i.e., a position for which the participants were expected to answer *yes*). Afterward, with a probability of 0.5, the end position was changed by selecting the nearest position downward the trajectory for which the correct answer was *no*.

For Task 2, it was important that a participant was able to determine the longest link. Therefore, some difference in length was required to distinguish the longest from the second longest link, allowing participants to perform Task 2. A threshold of at least 8% length difference as used, which was obtained from a preliminary study (see Section 5.3.5).

For both tasks, two different data densities were used: D1 and D2, see Table 5.2. The total number of points for D1 was 90, distributed across three trajectories each containing 30 points. More than one trajectory was necessary to simulate an appropriate use case. For D2, the number of points on each trajectory was doubled to 60, resulting in a total number of 180 points.

In Task 3, participants should recognize the cluster containing the most GPS locations. In the preliminary study, a threshold of 15% in the difference of the number of locations between clusters was determined. Therefore, the data set was generated so that the largest cluster had 15% more nodes than the second largest cluster, leading to a total number of 654 points distributed over 12 clusters. Each cluster had a radius of 200 m and a buffer zone of 50 m, to prevent visual merging of clusters. Within each cluster, the trajectories were generated with the Markov chain model. The overall trajectory was constructed by connecting clusters in random order.

5.3.5 Parameter Choices and Pilot Study

An informal small-scale study was conducted to set up the details of the parameters for the main study, in particular, the choice of thresholds discussed above. In this preliminary study, there were ten volunteers who were different from the participants of the main study. Eye tracking was not used in the preliminary study.

Participants were asked to select the longest link within five different stimuli, shown in random order, to set the threshold for Task 2. The stimuli had a varying difference in length between the longest link and the second longest link. The following thresholds were used: 2%, 4%, 6%, 8%, and 10%. Stimuli were rendered using NSP for point indication, RND for link sorting, and N (standard lines without direction cues) as link type. There were three randomly chosen trajectories with 30 to 60 nodes; this range of densities was used because it corresponds to the interval of densities defined by D1 and D2. At a threshold of 8%, 9 of 10 participants provided correct answers. Therefore, is can be assumed that this threshold is sufficient to allow for a fair identification of the longest link.

Participants were instructed to identify the cluster with the largest number of elements to determine the threshold for Task 3. There were seven different test data sets with varying difference in the number of elements between the largest and the second largest cluster, ranging from 3% to 21% in steps of 3% difference. Both NSP and SP were used for node rendering and N for link rendering. For a difference of 15%, 9 of 10 participants provided correct answers for NSP rendering and 8 of 10 for SP rendering. Therefore, is can be assumed that this threshold is sufficient to allow for a fair identification of the largest cluster.

A pilot eye-tracking study with another four participants was conducted before the main experiment to refine the final study design; in particular, the results showed that the number of repetitions for each participant had to be reduced to keep the average duration just above one hour per participant. In detail, the number of trials of Task 2 was reduced per block from 40 to 20, and for Task 1 from 40 to 16. Since Task 1 was more complex than Task 2, the completion time of a trial of Task 1 was higher than that of Task 2. Therefore, the trials per block of Task 1 were adjusted to achieve an overall completion time comparable to Task 2; in this way, participants spent approximately the same amount of time using the same link type in Tasks 1 and 2. The number of trails per block of Task 3 was not reduced since the completion time was not as high as for Tasks 1 and 2. See Table 5.3 for a detailed overview.

5.3.6 Study Procedure

The within-subjects design comes with a combination of the following factors: link type (A, EA, C, T, N), depth sorting (RND, SF, LF), node visualization technique (SP, NSP), and density of the data set (D1, D2). The concrete combi-

Table 5.3 — Summary of used methods and multipliers, describing the number of trials.

	Task 1		Task 2		Task 3	
Factor	Method	Multiplier	Method	Multiplier	Method	Multiplier
Depth sorting	(RND, SF)	2	(RND, LF)	2	(SF)	1
Density	(D1, D2)	2	(D1, D2)	2	-	1
Node rendering	(SP)	1	(SP)	1	(SP, NSP)	2
Repetitions		4		5		20
Trials per block		16		20		40
Link rendering	(A, EA, C, T)	4	(A, EA, C, T)	4	(N)	1
Trials per task		64		80		40
Participants	25					
Trials per task		1600		2000		1000

nation of factors depends on the task. Table 5.3 summarizes the relationship between factors and tasks. For Task 1, the depth sorting conditions RND and SF were tested, but not LF, since SF is more helpful for this path following task than LF, especially in dense regions, where the participants have to follow mostly short links. In contrast, RND was tested against LF for Task 2, where a good perception of long links is obviously beneficial. Link ordering SF was used as depth ordering for Task 3 because it shows best-detailed structures of clusters represented by short links. Furthermore, link rendering N was used, since it is the most *neutral* link type; thus, avoiding that the recorded data for NSP or SP as point indication is influenced by the properties of a specific link type.

The study was split into three blocks: one block for each task. In this way, the participants could adapt to the different tasks, and cognitive load from frequent context changes was avoided. At the beginning of each task block, the participants were given detailed information about the task and instructions; they performed exercises on a printed version, followed by a practice phase in front of the screen. The order of the task blocks was counterbalanced between the subjects to compensate for learning effects and fatigue.

For Task 1 and Task 2, there were four internal blocks of link rendering techniques (A, EA, C, T). Again, the order of these internal blocks was counterbalanced between the subjects. Within each internal block, 16 trials (for Task 1) or 20 trials (for Task 2) were performed. Here, the order of depth sorting (RND, SF or LF), data density (D1, D2), and the repetitions of these combinations were randomized.

Task 3 only evaluated the effect of node splatting (SP, NSP). The rendering technique was blocked. The order of the internal blocks was counterbalanced between subjects. The 20 repetitions within each internal block were randomized.

Participants were asked to fill in a questionnaire after each task to obtain subjective feedback on the respective task and techniques. The average duration of the study was about 70 minutes per participant.

5.4 Results

In this section, the results of the statistical analysis of average completion time and correctness for all tasks are presented. Time was measured in milliseconds; a log transformation was applied before statistical inference to conform to the normality assumption. Diagrams with timings are shown in units of seconds.

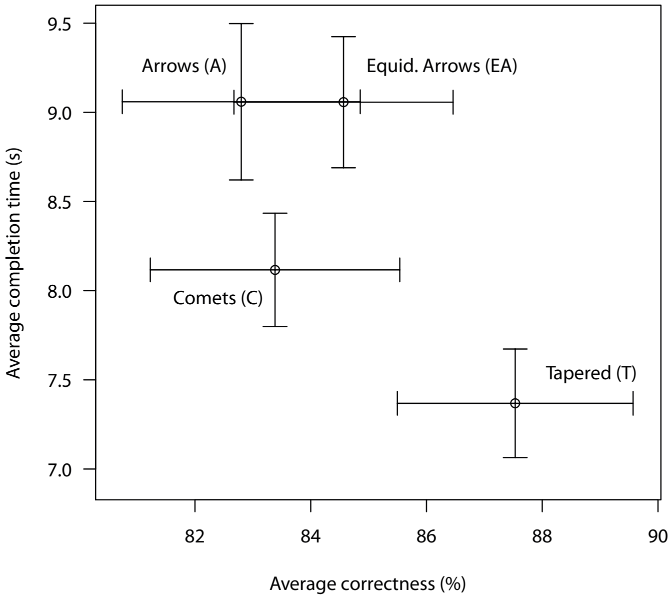
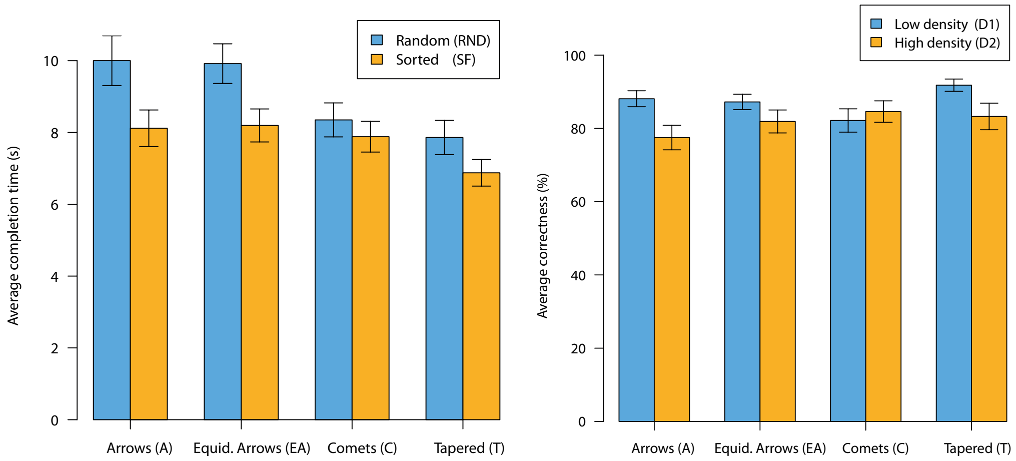


Figure 5.8 — Average correctness (x-axis) and average completion time (y-axis) for Task 1. Error bars show the standard error of means (SEM).



(a) Average time (b) Average correctness

Figure 5.9 — Performance for Task 1 and the different link types: (a) average completion time with different sorting options, (b) average correctness for different data densities. Error bars show the standard error of means (SEM).

In the following, abbreviations will be used for the different rendering methods and data set sizes, as summarized in Tables 5.1 and 5.2. The results of the statistical analysis of completion time and correctness are discussed for each task separately and are reported according to the description in Section 2.3.

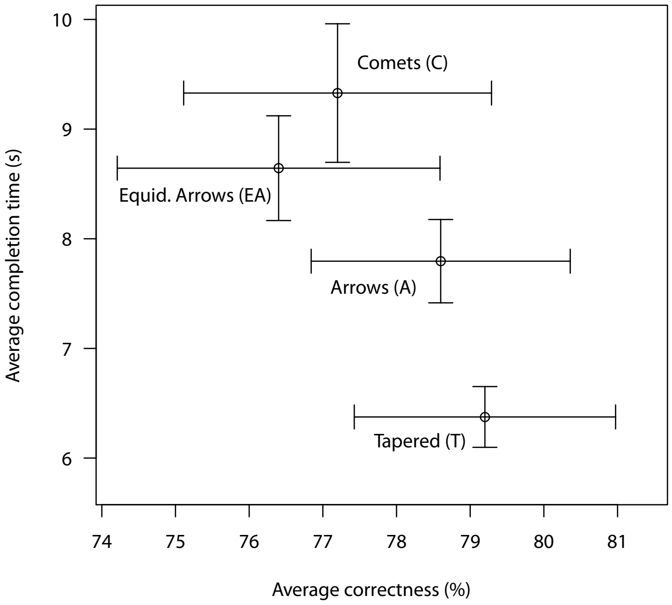
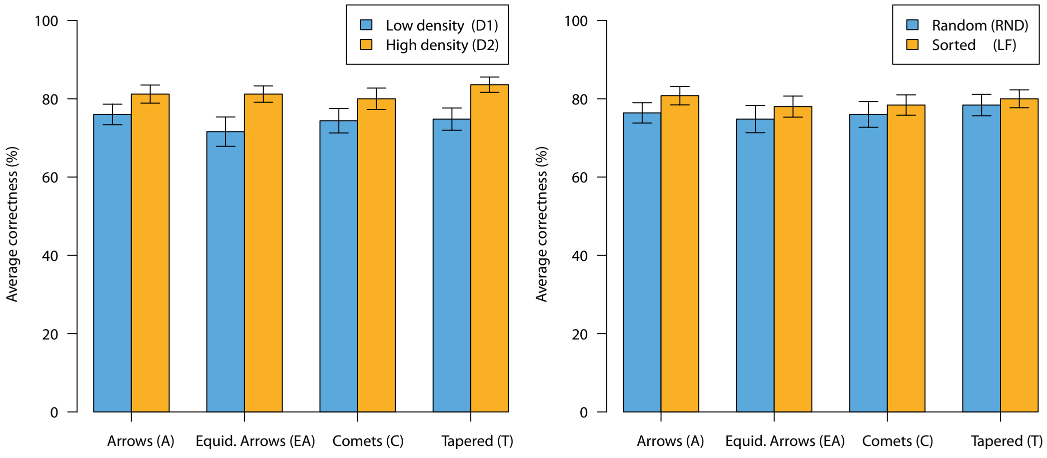


Figure 5.10 — Average correctness (x-axis) and average completion time (y-axis) for Task 2. Error bars show the standard error of the means (SEM).

Results for Task 1: With a mean correctness of 87.53% and a mean completion time of 7.37 s, T performed best. This is consistent with the results of Holten et al. [124]. C performed better than A and EA with respect to completion time, with a mean time of 8.12 s, and also better than A regarding the correctness with 83.38%. Figure 5.8 provides a summary of the descriptive statistical graphics of completion time and accuracy for the four link visualization techniques. ANOVA shows a significant effect of the link type ($F(3,72) = 9.45; p < 0.001; \eta_p^2 = 0.28$) and depth sorting ($F(1,24) = 22.27; p < 0.001; \eta_p^2 = 0.48$), concerning completion time. The post-hoc pairwise comparisons show significance for T compared to all other types ($p < 0.001$ for T-A and T-EA; $p = 0.046$ for T-C), and also between C and EA ($p = 0.011$). The influence of the depth sorting, leading to a decrease in time for each link type, is illustrated in Figure 5.9a. The correctness analysis exhibited only a significance for the data set density ($\chi^2(1) = 13.5; p < 0.001$). Here, the correctness decreases for all link types, excluding C, see Figure 5.9b. This indicates that it is slightly easier to handle C in regions with high density.

Results for Task 2: T performs best with an average completion time of 6.38 s and an average correctness of 79.2%, followed by A with 7.79 s and 78.6%. On average, participants needed the longest time (9.33 s) for C with a correctness of 77.2%, which is still better than the correctness of EA (see Figure 5.10). ANOVA shows significance for the link types concerning the completion time ($F(3,72) = 6.82; p < 0.001; \eta_p^2 = 0.22$). The post-hoc comparison exhibits signifi-



(a) Different densities (b) Different sorting

Figure 5.11 — Average correctness for Task 2 and the different link types: (a) with different densities, (b) with different sorting methods. Error bars show the standard error of means (SEM).

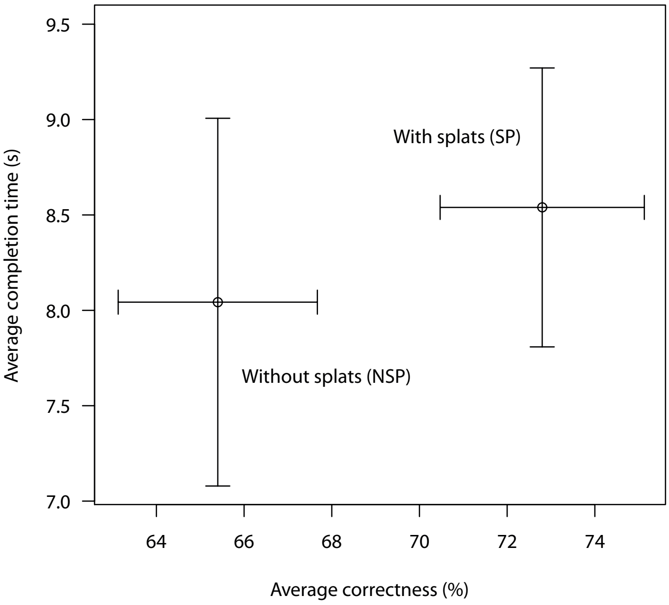


Figure 5.12 — Average correctness (x-axis) and average completion time (y-axis) for Task 3. Error bars show the standard error of the means (SEM).

cant differences between T and all other types ($p < 0.001$), and also between C and A ($p = 0.012$). For correctness, there is a significance for the data density ($\chi^2(1) = 11.64; p < 0.001$) and depth sorting ($\chi^2(1) = 5.2609; p = 0.022$). Figure 5.11a illustrates that the correctness increases with the density summarized over all sorting options. Figure 5.11b shows that sorting also affects the correctness summarized over all densities.

Results for Task 3: The analysis showed a significant effect for the correctness ($\chi^2(1) = 4.17; p = 0.041$), with an average correctness of 72.8% for SP and 65.4% for NSP; see Figure 5.12. While SP achieves a better correctness, it comes with a higher average completion time of 8.54 s compared to NSP with 8.04 s (not significant).

5.5 Eye-Tracking Evaluation

In this section, the results of the eye-tracking measurements are presented. A statistical analysis of the average fixation duration and average saccade length was performed per displayed image since these are common metrics in eye tracking.

The perceptual and cognitive interpretation of these eye-tracking metrics should be used with caution because of the complex perceptual and cognitive processes involved. However, these metrics add a useful view on this user study and allow in this case an interpretation of visual stress as discussed in Section 2.2.2.

A common characteristic of eye-tracking metrics, including the ones employed, is that they do not exhibit normal distribution. Therefore, non-parametric significance tests are required.

Eye-Tracking Evaluation for Task 1: Figure 5.13 shows an overview of duration and saccade length of all link types. C has here both the longest average saccade length of 130.62 pixels and the highest average fixation duration of 940.15 ms. For EA, the smallest values for these parameters are obtained: 115.16 pixels and 773.41 ms. The statistical analysis shows a significance in duration for the density ($\chi^2(1) = 25; p < 0.001$). This influence—increasing duration with increasing data set size—is shown in more detail in Figure 5.14a. For the saccade length, there is a significance for the density ($\chi^2(1) = 6.67; p = 0.009$). A decline of the saccade length is depicted in Figure 5.14b. In both cases, this is consistent with the expectations that a higher data density would lead to a higher average duration and a reduced average saccade length, since more complex images may need more attention (longer duration) and have to be investigated in more detail (shorter saccades).

Eye-Tracking Evaluation for Task 2: In this task, again, the smallest values for the average duration and saccade length were measured for EA (216.46 pixels and 287.97 ms). The longest duration was obtained for C (220.06 pixels with 304.97 ms) and the longest saccade length for T (242.96 pixels with 291.94 ms). Figure 5.15 shows an overview of the eye tracking metrics. The statistical analysis showed only significance for the average saccade length. Here, all three

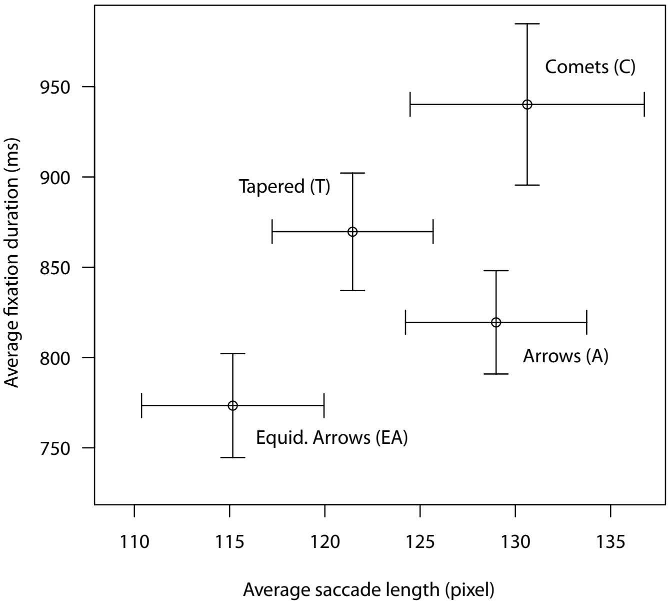
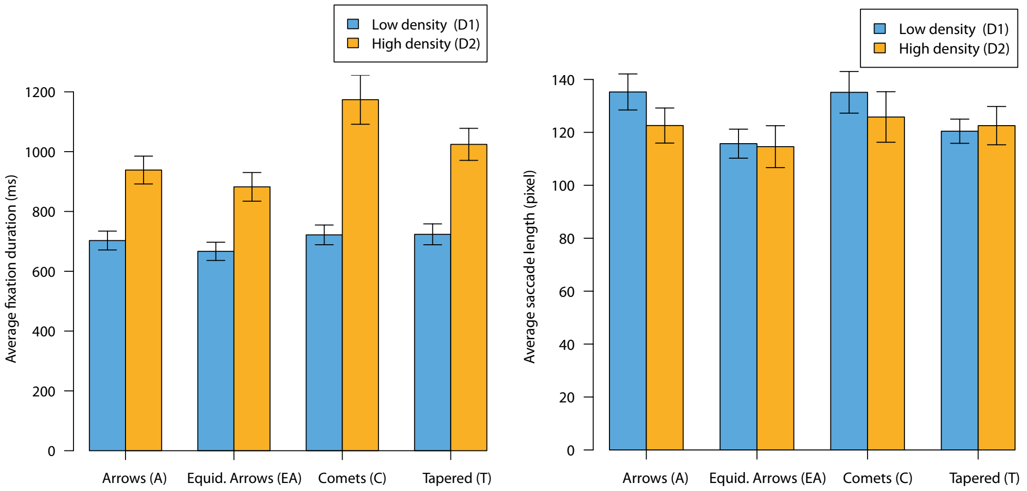


Figure 5.13 — Average saccade length (x-axis) and average fixation duration (y-axis) for Task 1. Error bars show the standard error of the means (SEM) for the eye tracking data.



(a) Average fixation duration (b) Average saccade length

Figure 5.14 — Average fixation duration and saccade length for each link type with different densities for Task 1. Error bars show the standard error of means (SEM).

factors were affected: link types ($\chi^2(3) = 35.35; p < 0.001$), density ($\chi^2(1) = 9; p = 0.003$), and depth sorting ($\chi^2(1) = 11.56; p < 0.001$).

The post-hoc analysis for the link types reveals significance between T-EA

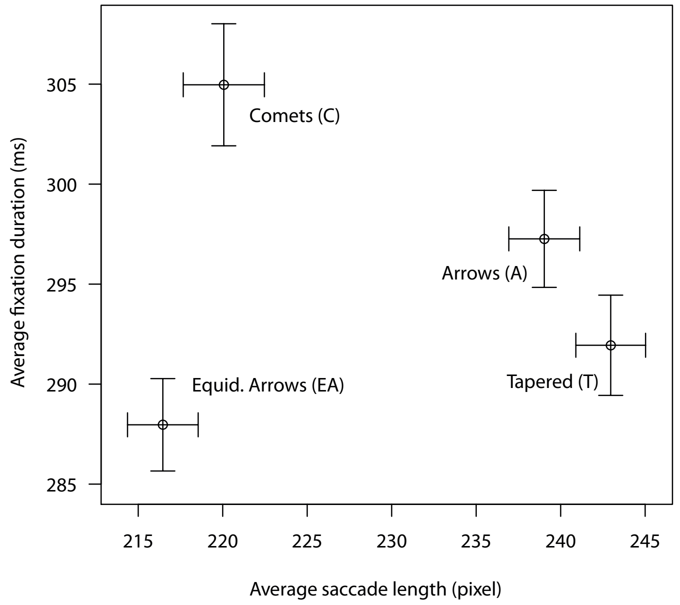
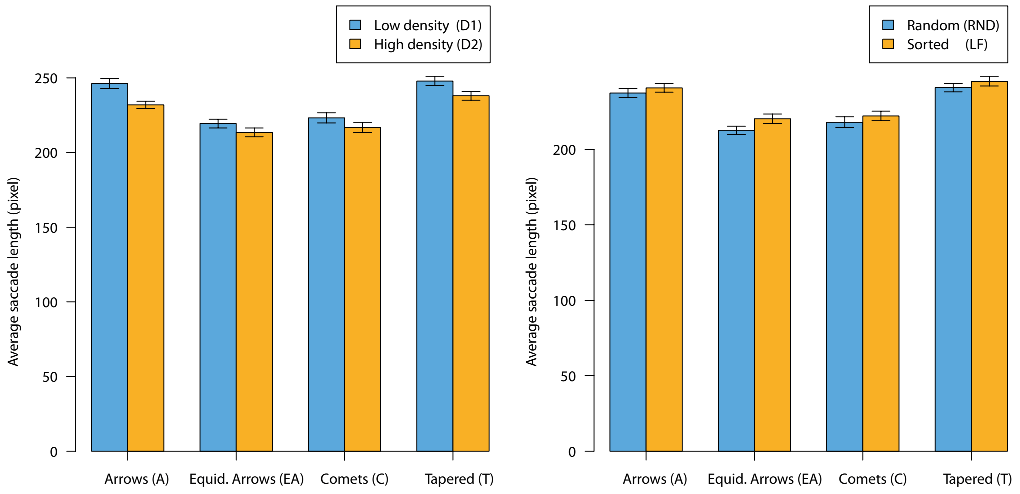


Figure 5.15 — Average saccade length (x-axis) and average fixation duration (y-axis) for Task 2. Error bars show the standard error of the means (SEM) for the eye tracking data.



(a) Different densities (b) Different sorting

Figure 5.16 — Average saccade length for Task 2 and the different link types: (a) with different densities, (b) with different sorting methods. Error bars show the standard error of means (SEM).

($p < 0.001$), T-C ($p < 0.001$), C-A ($p = 0.003$), and EA-A ($p < 0.001$). The influence of density and sorting for each link type is shown in Figures 5.16a and 5.16b. According to these plots, the average saccade length decreases with increasing density, while the sorting has the opposite effect. This may be interpreted as

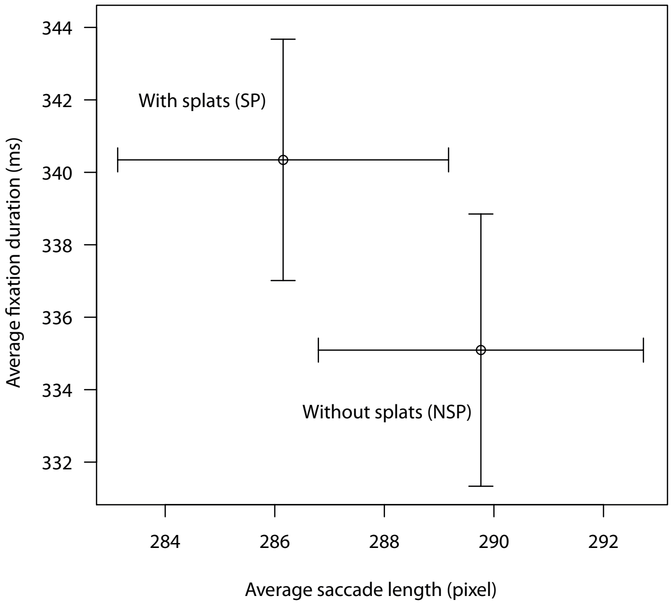


Figure 5.17 — Average saccade length (x-axis) and average fixation duration (y-axis) of Task 3. Error bars show the standard error of the means (SEM) for the eye tracking data.

follows: increasing the data density also increases the visual stress resulting in shorter saccades, whereas the opposite holds for the sorting.

Eye Tracking Evaluation for Task 3: On average, NSP achieved a slightly shorter fixation duration of 335.09 ms and also a slightly longer saccade length of 289.76 pixels compared to SP with 340.34 ms and 286.15 pixels. This might vaguely indicate that participants needed more attention investigating clusters while SP was used. However, the statistical analysis showed no significance. Figure 5.17 provides the respective statistical graphics.

5.6 Discussion

In this study, the *statistical evaluation* block of the workflow in Figure II consists of a traditional task performance evaluation. Additionally, eye-tracking metrics were used to show characteristics of eye-movement behavior of participants for the different directional encoding methods.

One result of this study is that T performed better than the other link types for Task 1, with the highest correctness and the lowest completion time (Figure 5.8), confirming H1. Furthermore, H3 could be confirmed: depth sorting led to a reduction of the solution time (Figure 5.9a). Also, the evaluation of the eye tracking data revealed that A had the highest average saccade length and the shortest average fixation duration. This may be interpreted as intuitive and efficient usage of A (Figure 5.13)—with less of short, hectic eye movement, exhibiting less visual stress compared to the other methods.

For Task 2, T again performed best, concerning the solution accuracy and time, as depicted in Figure 5.10. Therefore, H2 was rejected. From Figure 5.10, one can see that group (C, EA) with repeating directional cues is outperformed by the techniques A and T, which come with only one cue (an arrow at the end or a prolonged arrow). Figure 5.15 shows that the techniques A and T have similar eye-tracking characteristics regarding average fixation duration and saccade length. T led to the longest average saccade length and the lowest fixation duration, compared to the other link types; this can be interpreted as an indicator of high efficiency. The high fixation duration and short saccade length for C and EA can be explained by the fact that participants counted the comets or arrows to estimate and compare the length of links. Furthermore, H4 was confirmed because sorting leads to higher accuracy, which is true for any link type (Figure 5.11b).

Using SP as point visualization in Task 3 reveals an increase of both accuracy and completion time compared to using NSP (Figure 5.12). The eye-tracking evaluation showed a lower average fixation duration and a longer saccade length for NSP (Figure 5.17). According to this result, participants might have investigated clusters more carefully when SP was used, which could explain the higher average solution correctness and increased solution time, shown in Figure 5.12. Since accuracy was valued higher than response time, it could be argued that the findings support H5.

Overall, T performed very well in all tasks, followed by A. This gives evidence that participants can handle simple link types quite well. Nevertheless, the results for solution accuracy and completion time for C and EA were similar to those of T and A. Also, the participants stated that they liked the equidistant comets. Therefore, the equidistant comets can be recommended for trajectory visualization in large and complex data sets. This reasoning is based on the observations described in Section 5.2.1 and illustrated in Figure 5.2: For such data sets, detailed exploration requires frequent zooming. However, zooming affects the readability of direction information in T and A. In fact, that information might be completely missing is such visualizations. In contrast, C and EA are not affected by zooming, which makes it possible to interpret the direction of a link independent of the zoom level. It should be pointed out that stimuli with zooming were not included because the disadvantages of T and A are obvious by construction, and it was not necessary to confirm them with this user study.

Chapter 6

Visual Search Support

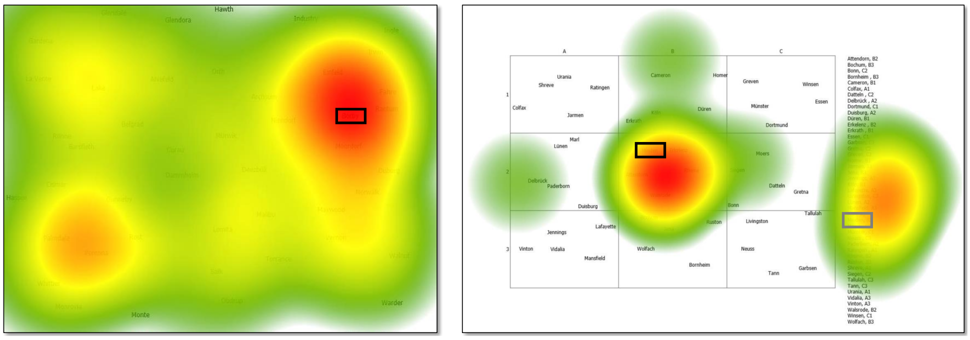
There are many tasks that require a visual search for objects in an image. Especially in the context of visualization, such search tasks are often supported by appropriate interaction methods: typically, the user can select an object type, and the respective positions are highlighted in the visualization. However, there are scenarios in which only static representations are available or interactive selection is not provided, e.g., in printed maps or in computer-based visualizations that do not facilitate appropriate highlighting.

The latter scenarios are evaluated in this chapter[1] in a study with 32 participants to understand visual cues that support visual search in 2D visualizations. Geographic maps were chosen as objects of the study because they are one of the oldest types of visualization, they are in wide use, and they serve as a prototypical example of 2D visualization. Instead of real-world maps, a simplified version of those was used: a 2D space with text labels corresponding to the objects that participants had to find visually. With this design, confounds from additional visual elements are avoided, focusing on the actual search task.

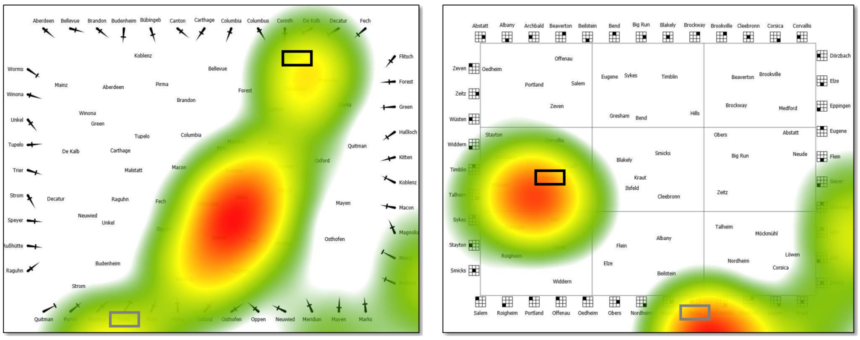
In a controlled eye-tracking study, four map annotation techniques were compared. The analysis of the eye-gaze data allows inference about possible gaze

---

[1] **Parts of this chapter have been published in:**
R. Netzel, M. Hlawatsch, M. Burch, S. Balakrishnan, H. Schmauder, and D. Weiskopf. An evaluation of visual search support in maps. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):421–430, 2017 [7], © 2017 IEEE.



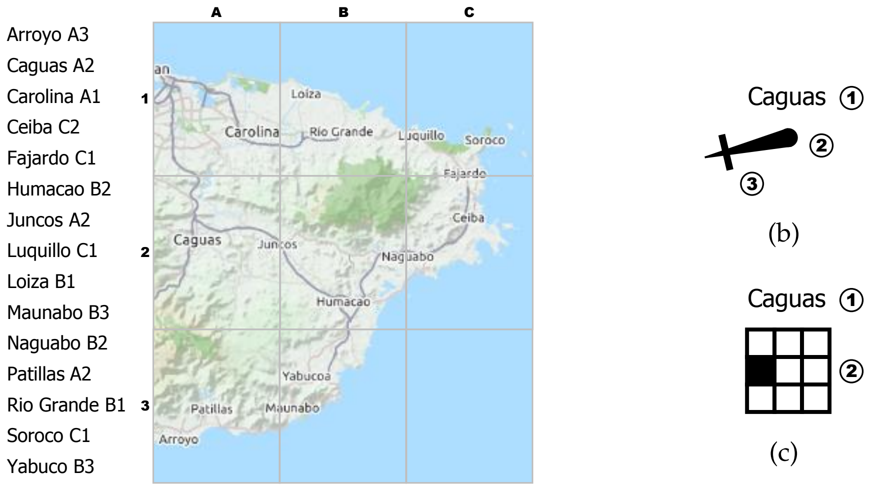
(a) Within-image annotation (WA) (b) Grid reference annotation (GA)



(c) Directional annotation (DA) (d) Miniature annotation (MA)

Figure 6.1 — Heatmaps of the eye-tracking data of one participant for the four evaluated variants of map annotation with labels. The different annotation variants lead to significantly different gaze characteristics. The target label is marked with a black box. In the case of the methods with support for visual search ((b)–(d)), the label is also marked with a gray box in the ordered list outside the map.

patterns, workload, and cognitive processes of participants. The baseline is *within-image annotation* (WA; Figure 6.1a), which contains the labels on the map without any support for visual search. Traditional *grid reference annotation* (GA; Figure 6.1b) is commonly used in city or street maps, enriching the map by an additional sorted list of labels that come with textual coordinates referring to cells in the map (e.g., grid cell "C4"). Other methods in this study utilize a sorted list of elements indicating the positions of the labels within the map to provide search support. With annotations, the search space on the map is heavily reduced. However, grid reference annotation requires switching between a textual representation of a sub-area of the map and its actual visual location on the map. Therefore, two new annotation variants were designed: *Miniature annotation* (MA; Figure 6.1d) avoids this mental switching by replacing the textual coordinates with a graphical index in the form of a miniature grid



(a)

(b)

(c)

Figure 6.2 — (a) Traditional grid reference as commonly used in maps. It combines a geographic and a list-based representation of the labels in alphabetical order. The labels are combined with grid reference descriptions to accelerate the label search task in the map. (b) Directional annotation uses an arrow (2) to indicate the direction in which the element with the respective label (1) is located. The distance to this element is shown with a line marker (3) on the arrow. (c) Miniature annotation uses a miniature grid (2) to indicate the location of the element with the respective label (1). The grid corresponds to the grid overlaid on the map and the cell that contains the element is marked.

that highlights the indexed cell. *Directional annotation* (DA; Figure 6.1c) uses a graphical index in combination with an arrow to point to the location of a point within a map. Therefore, it reduces the search space to a straight line (in the direction of the arrow), instead of reducing it to an area, e.g., a cell as it is the case for grid reference or miniature annotation. For miniature and directional annotation, the supporting labels and graphical cues are placed at the border of the map. All four map variants are displayed in Figure 6.1, overlaid with representative examples of gaze data from the study.

One result of this eye-tracking study is that maps with supporting annotations lead to better search performance than the baseline within-image annotation. Best task completion times are achieved with the miniature annotation approach. The analysis of the eye-movement data reveals interesting aspects of visual task solution strategies: participants apply significantly different reading strategies for the different annotation methods; different steps were identified that participants performed during the search task.

6.1 Related Work

Visual search is studied in many fields, ranging from perceptual psychology all the way to its application in interactive visualization [157, 158, 159, 128, 160]. These studies either investigate if visualization is useful for a given task or they focus on how visualizations can be enhanced concerning visual cognition, perception, and attention.

A typical example of improving the visual search is the use of grid references (see Figure 6.2): the map is visually enhanced by an (alphabetically) ordered list of the text labels (such as street names) associated with a textual coordinate of a grid cell (e.g., "B2"). With this approach, visual search is still required but is more efficient, since it is restricted to a certain cell. Many researchers argue that grids are beneficial compared to other reference line systems because grids are simple, more accurate, and facilitate map compilation [161].

Although grid references are useful and support the guidance to a smaller region in which the target is located, a viewer first needs to build the correspondence between the textual description of the grid cell (coordinates) and its spatial location, which comes with an additional cognitive load.

An intuitive way of establishing visual links is by drawing lines that connect two elements, e.g., as in node-link diagrams. Such diagrams typically utilize full-length straight links that perfectly guide the viewer from one location to the connected object [124]. This approach can also be used to connect boundary labels around maps with interior points [162]. Unfortunately, diagrams with many full-length links can exhibit visual clutter [163]. Therefore, crowded maps do not work with standard link representations; however, partial links [133] that point into the coarse direction of a location are a promising substitute. Alternatively, directional information could also be shown by partial circles whose curvature, respectively size, indicates distance [164]. Another method that still connects labels at the border of the map with the actual position in the map with a line is presented by Kindermann et al. [165]. They rearrange the labels at the border and bend lines to avoid crossings and clutter. As another variant, miniature annotation was used. Here, correspondence is not shown explicitly as for direct links, it is rather given implicitly by marking the corresponding grid cell in a simplified miniature representation of the map domain. Miniature representations are often used in visualization, e.g., for sparklines [166] or as 3D miniature maps to support navigation in 3D virtual worlds [167]. Another example is TimeRadarTrees [168], which use a radial grid miniature annotation to visualize the correspondence to a large context representation of a dynamic graph. Furthermore, origin–destination maps can be visualized in a way that miniature versions of the map are placed inside the cells of a regular grid [169], which is similar to a 2D dimensional stacking (see Section 4.4.1). In this study, the design of the used miniature representation is also grid-based, but grid cells are not further subdivided since only one location is associated with a miniature representation. For map design in general, there are several suggestions, guidelines, and principles; see, e.g., the books by Robinson [170], Robinson et al. [171], or Tyner [172]. According to Tyner, the design goals are clarity, order, balance, contrast, unity, and harmony. Our annotation approaches do not change the content of carefully designed maps, but rather include small additional information outside of the map to support a reader in performing a search task. There are other examples of enhanced map visualization based on further graphical elements, including label placement for metro maps [173], map labeling [174], graphical legends [175], highlighting of interesting locations in tourist brochures [176], or annotation with loop lines [177]. Loosely related, there is a study of participants' performance (latency and accuracy) for tasks with three different map legend designs [178]. Legends are used here to provide the meaning of a symbol on the map. Their findings show no best or least efficient map legend, but the authors concluded that the design should be influenced by the purpose of the map and also by the opinion of users. Another eye-tracking study that was presented in Chapter 5 of this thesis assesses the readability of links encoding directional information in trajectory visualization. Here, eye-tracking metrics are used to get insight into cognitive processes of participants.

Independent of which map or annotation is used, a viewer is supposed to establish a correspondence between the available location information and the actual position on the map. Therefore, the topic of the general problem of correspondence visualization is of interest, which also includes visual comparison (e.g., for multidimensional data [179]) or visual support in multiple coordinated views [180, 117]. The concepts and research results in these fields can be applied to the visual search support provided by map annotations. Here, annotations provide perceptional cues to make relations between labels and actual locations in the map clearer and focus the attention of participants on specific areas on the map.

6.2 Visual Search Support

To support search tasks in maps, a connection between an alphabetically ordered list of elements and their position in the map has to be provided. Thereby, the reader first searches the ordered list for the element of interest. The list provides then a hint about the whereabouts of the element on the map.

6.2.1 Within-Image Annotation

The baseline for comparison is a map with labels directly placed at their corresponding positions in the display space (Figures 6.1a and 6.4a). This within-image annotation does not provide any extra visual cues to accelerate the search for labels on a map. In the worst case, the reader has to scan the entire map. Furthermore, there is no inherent ordering of labels, like in an alphabetically ordered list, i.e., no additional support for a faster search.

6.2.2 Grid Reference Annotation

The traditional grid reference approach is commonly used in maps (Figure 6.2a). A 2D Cartesian coordinate system divides the map into smaller cells. An alphabetically ordered list provides an index for the names and labels contained in the map. This list includes respective textual coordinates for each entry (e.g., *B2*). The reader then only has to search in the cell with the coordinates provided by the list (Figures 6.1b and 6.4b).

Based on this concept, two variants of annotation techniques were designed that support search tasks in maps: directional annotation and miniature annotation. Both techniques have in common that they use a more direct visual connection between the ordered list and the map. In contrast, the traditional grid reference uses two coordinates in a textual representation to indicate the respective cell in the map.

6.2.3 Directional Annotation

This annotation technique uses an arrow to point in the direction of the element of interest (see Figure 6.2b). The shape of the arrow resembles a tapered link, which was shown to be an efficient link representation [124]. Instead of the full link length, a shortened version was used in the sense of a partial link [133]. With this approach, the user has to search for the element in the indicated direction. To further improve the search, a marker on the arrow shows the distance of the annotation to the element. Hence, this annotation technique uses direction and distance as cues for the location of the element of interest.

There are other possible encodings for the direction and distance of an element, e.g., the distance could be mapped to color. However, this design is more intuitive because a direct representation of both quantities was used, i.e., direction is represented by the angle of the arrow and distance is represented by the position of the marker.

With this approach, the duration for finding a designated element depends on the proximity of the annotation to the element. The ordered list of elements is placed directly around the map to minimize the average distance between arrows and labels within the map (Figures 6.1c and 6.4c).

6.2.4 Miniature Annotation

Miniature annotation is more closely related to the traditional grid reference concept. A grid is used to divide the map into different cells. Instead of using a 2D coordinate system to indicate the cell with the designated element, a visual miniature representation of the grid is used where the respective cell containing the label is marked (see Figure 6.2c). With this approach, it is not required to read the textual coordinates and search for them along the coordinate axes of the map.

This approach has some advantages compared to directional annotation. Instead of using two different encodings for direction and distance, the miniature representation incorporates both. This might be more intuitive to understand. Furthermore, when using the directional annotation without any additional visual aids, the reading accuracy decreases with the distance of the element [181]. In contrast, the reading accuracy of the miniature annotation is independent of the location of the respective element. Finally, it is not required to place the miniature annotations close to the map.

A drawback of this approach is that the location of the designated element is only coarsely represented, while the directional annotation points to the exact location. The accuracy of the directional annotation only depends on perceptual accuracy, but not on the resolution of the miniature discretization.

One design parameter for the miniature annotation is the number of cells of the used grid. With a small number of cells, the marked cell can be identified faster, e.g., with a 3×3 grid (see Figures 6.1d and 6.4d), it is only required to see if the marked cell is in a corner or in between. However, the size of the cells is large in such cases and contains many elements that must be scanned. With a higher number of cells, fewer elements are contained. Thus, the identification of the row and column position of the marked cell requires more time. The same issue occurs with the traditional grid reference annotation. For grids with a very large number of cells, the traditional approach might perform better since the coordinates of the respective cell are directly shown.

6.3 Model of Visual Search in Annotated Maps

In this section, idealized steps are discussed that are necessary to read the different annotations. These steps will build the basis to discuss the scalability of

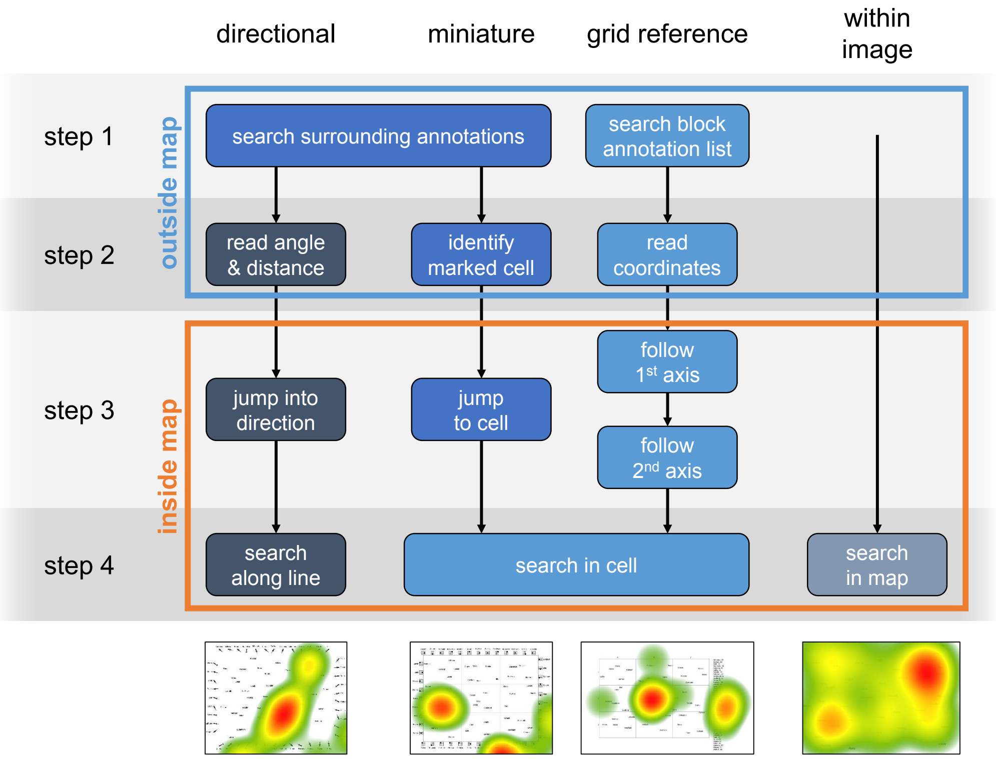


Figure 6.3 — Steps that have to be performed with the annotation methods. These steps are reflected in the results of the eye-tracking study (heat maps at the bottom, for one participant).

the variants (Section 6.6) and lead to hypotheses about user performance as well as reading and visual task solution strategies with the respective visualizations (Section 6.4.1).

The different annotation approaches require different steps that have to be performed to find an element of interest in the map (Figure 6.3). The annotation variants (DA, MA) were designed to improve some of these steps in comparison to the traditional grid reference approach.

Step 1: The first step describes the search for the label of the element of interest. Here, the labels are ordered alphabetically for rapid identification. The directional and miniature annotation place graphical representations around the map in clockwise order, whereas the grid reference uses a more compact text block. Therefore, the search with the grid reference is restricted to a smaller area and possibly faster. However, with the directional and miniature approaches, the annotations are closer to the map, which can be beneficial for the subsequent search steps.

Step 2: In the second step, the search cue in the annotation needs to be recognized. The directional annotation requires the reader to estimate the distance to the target using the distance marker before the search can start in direction of the arrow. In the case of the miniature annotation, the marked grid cell must be identified, whereas the grid reference uses textual coordinates for this purpose. It is assumed that this step can easily be performed with the miniature annotation, at least for the selected size of the grid (3×3), since it uses a direct visual representation of the auxiliary grid on the map.

Step 3: In step three, the search for the area, indicated by the annotation, is performed. Miniature and grid reference annotation differ here in the way how they indicate the respective cell of the auxiliary grid. Is is assumed that this step is faster with miniature annotation, since the reader can directly jump to the marked cell, whereas the grid reference requires additional look-ups at the two axes of the coordinate system. In the case of the directional annotation, the duration of this step depends on the reader, since there is no visual feedback if the correct area has been found. If the reader performs this step fast but inaccurately, it will take longer to find the correct element in the subsequent search step.

Step 4: The last step describes the final search for the element of interest in the search area indicated by the annotation. In the case of the within-image annotation, this is the only performed step, since no additional search support outside the map is provided; the search area is not reduced and therefore the search must be performed in the full map. The miniature and the grid reference annotations benefit from the auxiliary grid on the map. In contrast, it is not feasible to combine the directional annotation with auxiliary lines, since many of them would intersect with each other in different angles leading to visual clutter. Therefore, the directional annotation provides a less exactly defined search area than the other two approaches. This might lead to a longer search process, especially in the case of larger distances between the annotation and the element of interest because the probability is higher that the reader leaves the search line defined by the arrow.

6.4 Experiment Design

A controlled laboratory eye-tracking study was conduced to evaluate and compare different annotation methods. Within-image annotation serves as baseline for the comparisons. The study investigates task performance in terms of completion times as well as characteristics of the recorded gaze data. In the following, the design and execution of the study are discussed.

6.4.1 Hypotheses

Based on the theoretical considerations from Section 6.3, two hypotheses were built with respect to task completion times before running the experiment:

H1: Without annotations, the search for certain textual elements requires more time, since all textual elements have to be read in the worst case. Therefore, longer average completion times are expected for within-image annotation than for stimuli produced with the other annotation methods (directional, miniature, and grid reference annotations). The following ordering of completion times is expected: $WA > GA$, $WA > DA$, $WA > MA$.

H2: There will be differences concerning the completion time between directional, miniature, and grid reference annotations. It is expected that $GA > DA > MA$.

Hypothesis H2 reflects that there are different perceptual sub-tasks involved with these annotations (see Section 6.3 and Figure 6.3).

Typically, the evaluation of error rates would be considered in task-performance studies. However, the pilot studies indicated that participants were always able to find the designated label reliably, and therefore, significant differences in task accuracy were not expected. The high accuracy is plausible because it is easy to check whether a text label is identical to a specified text, once the label is found.

The following hypotheses concern gaze characteristics and reading strategies that can be inferred from eye tracking. The reading strategies are formulated according to Figure 6.3. To this end, areas of interest (AOIs) are defined for the outer regions of the stimuli containing the annotations (*outside*) and the interior area where the labels are located (*inside*). The AOIs are illustrated in Figure 6.4. Steps 1 and 2 of Figure 6.3 take place in *outside*, steps 3 and 4 in *inside*.

H3: The average saccade length for directional, miniature, and grid reference annotations is larger than for within-image annotation because the ordered list of labels and the search cues allow longer jumps within the list and to the target.

H4: For directional, miniature, and grid reference annotations, the visual search starts in the outer area (*outside*), proceeds then to the interior (*inside*), and ends there at the target. The average fixation duration for *outside* is expected to be smaller than for *inside* because the actual search for a label requires more attention than utilizing the visual aid to estimate the rough position of the label.

H5: In the interior (*inside*), the gaze movement patterns for directional annotation are different from miniature and grid reference annotations. The gaze is expected to be located along a search line for directional annotation. In contrast, participants should search in the rectangular area of the cell while using miniature or grid reference annotations. Therefore, the angular differences between subsequent saccades are expected to be lower for directional annotation (i.e., saccades along a line) than for the other two types of annotation.

6.4.2 Stimuli and Task

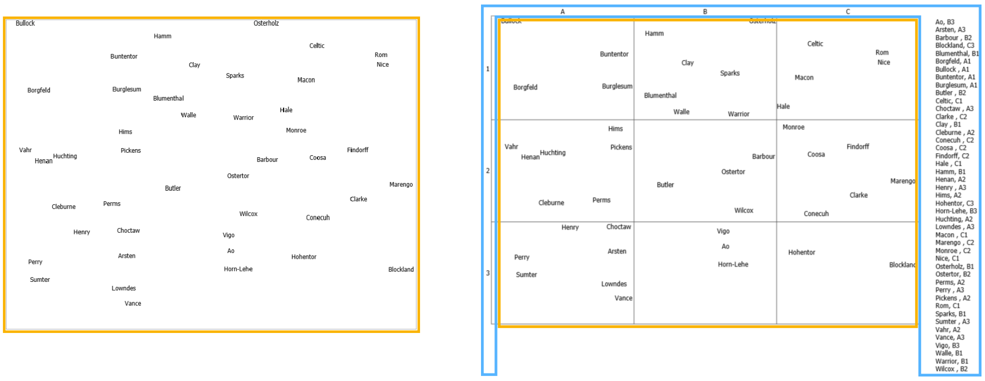
Artificial datasets were generated to avoid that participants would have prior knowledge about a map used in the study. The maps contained the names of major cities of the US, France, Germany, and the United Kingdom. Each label was displayed with the same text representation parameters, i.e., font size, font face, text orientation, and color. A grid structure of 3×3 was utilized to distribute the labels within the stimuli. In each grid cell, five different labels were randomly placed, leading to 45 labels per stimulus to achieve an initially equal distribution of labels. Up to three labels were removed or added randomly to each cell to avoid a too regular distribution. This process did not change the total number of 45 labels.

The size and position of the map area were identical for all stimuli. The generated images had a size of 1000×900 pixels. Figure 6.4 and 6.8 shows examples of the stimuli.

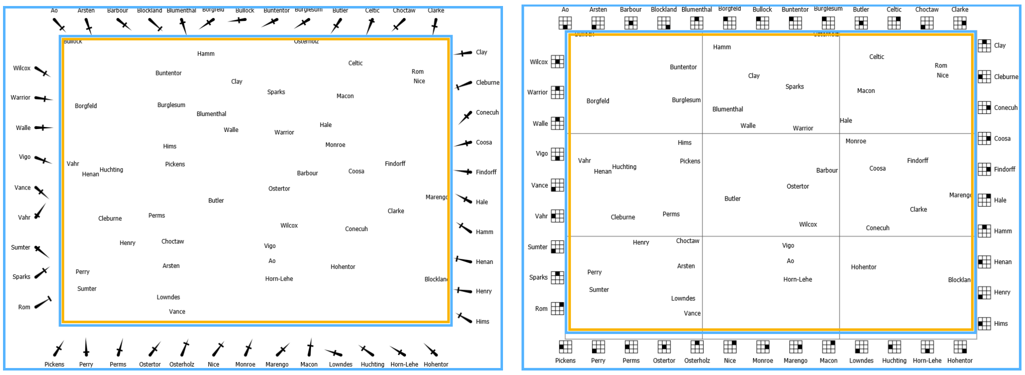
The task that the participants had to perform was as follows: *Find the specified label within the map*. The label was shown to the participants before the stimulus was shown, and was varying from stimulus to stimulus. The participants were asked to perform the task as fast and accurately as possible, but with the focus being on high accuracy.

6.4.3 Study Procedure

Overall, 80 maps were generated for each of the four methods, resulting in a total number of 320 stimuli. These numbers were adjusted based on the results of a pilot study, to meet the time constraints for the whole procedure. A subset



(a) Within-image annotation (WA) (b) Grid reference annotation (GA)



(c) Directional annotation (DA) (d) Miniature annotation (MA)

Figure 6.4 — The rectangles indicate the AOIs used for the analysis of the eye-tracking data (blue for *outside* and orange for *inside*).

of these stimuli was presented to each participant in a way that each stimulus was read at least by eight participants.

Overall, a participant was confronted with 80 stimuli (20 for each visualization method). A two-staged counterbalancing was employed to compensate for learning and fatigue effects. By using two stages, the counterbalancing can be achieved with a smaller number of participants. First, the 80 stimuli of one participant were divided into four blocks each consisting of 20 images, containing visualizations from all four methods. The block order was counterbalanced. Second, each block was further subdivided into four sub-blocks (in counterbalanced order), each containing five stimuli generated with the same visualization technique. Their order within each sub-block was randomized.

During the task execution for the study, the presentation of a stimulus was decoupled into two parts: first, only the name of the target label was presented, and after pressing a key, the stimulus itself. The subjects were not allowed to use auxiliary means, like the mouse or their fingers, during the search for the

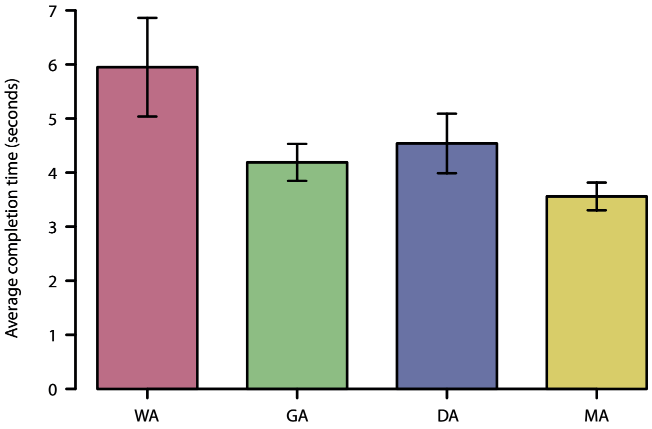


Figure 6.5 — Completion times for within-image annotation (WA), grid reference annotation (GA), directional annotation (DA), and miniature annotation (MA): average times along with error bars that show the standard error of the means (SEM).

correct target, since this could affect the performance of directional annotation. After pressing a key, all measurements were stopped, and participants were able to select the found target using the mouse. Afterward, the next stimulus was presented.

6.5 Evaluation

This section presents the statistical evaluation of task performance, the analysis of the recorded eye-tracking data, and the results of the subjective feedback obtained through the questionnaire. Results are presented as described in Section 2.3.

6.5.1 Task Performance

The evaluation was performed on task performance based on completion times. The best results were achieved with miniature annotation (MA) followed by grid reference annotation (GA), directional annotation (DA), and within-image annotation (WA) at last. The participants performed the task on average in 3.56 s (MA), 4.19 s (GA), 4.54 s (DA), and 5.95 s (WA), see Figure 6.5. The average task completion time overall techniques were 4.56 s. Expressed in percentage compared to the baseline WA, MA was 40.2% faster, GA 29.6%, and DA 23.7%. The statistical test revealed significant differences between some of the methods ($\chi^2(3) = 45.42; p < 0.001$). The results of the post-hoc analysis are summarized in Table 6.1.

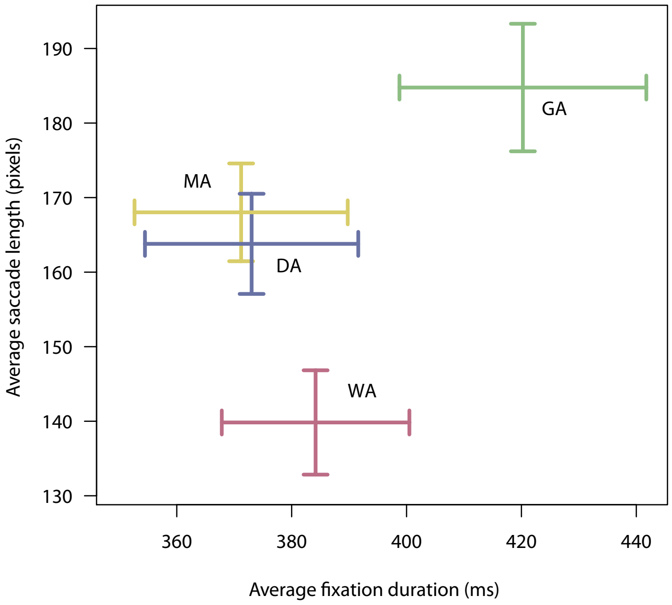


Figure 6.6 — Average fixation duration (x-axis) and average saccade length (y-axis) for all four annotation methods. Error bars show the standard error of the means (SEM) for the eye-tracking data.

There are significant differences between WA and all other annotation methods ($p < 0.001$). Furthermore, significant results can be found between MA-GA ($p = 0.01$) and MA-DA ($p < 0.001$). There are no significant differences between DA and GA.

These findings confirm hypothesis H1: As expected, visual search support improves task performance compared to the baseline WA. However, hypothesis H2 can only be accepted partially, because there is no significant difference between GA and DA, but MA exhibits better performance than both.

6.5.2 Eye-Tracking Data Analysis

In addition to a traditional task performance analysis, gaze data are used for a more detailed evaluation with inference of, e.g., visual interaction patterns

Table 6.1 — P-values of post-hoc comparisons of completion times.

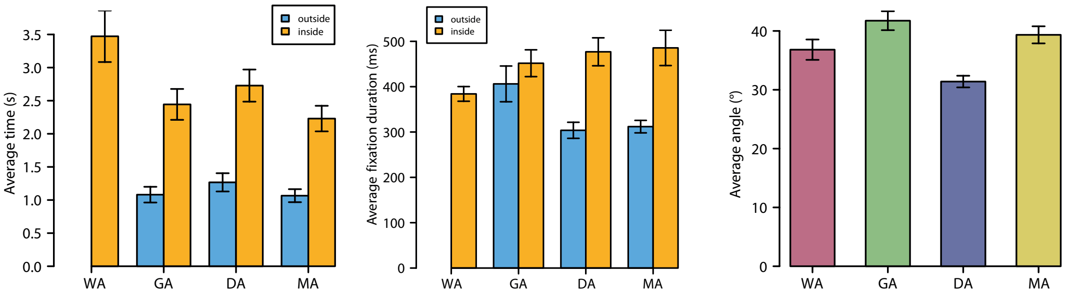
Method	GA	MA	DA
MA	0.01	-	-
DA	0.26	<0.001	-
WA	<0.001	<0.001	<0.001

of subjects, their workload, or cognitive processes while they were solving a task. The first point is often based on the evaluation of AOI sequences, while the latter two utilize different metrics for quantification.

To this end, the average fixation duration and average saccade length are utilized (see Section 2.2.2) in a similar fashion as in Chapter 5. Additionally, AOIs are used for the evaluations of the study described in this chapter.

The eye-tracking results are being analyzed with a statistical evaluation of average fixation duration and average saccade length as dependent quantities. Figure 6.6 summarizes the results. ANOVA indicated that there is a significant impact of annotation type on average fixation duration ($F(3,116) = 5.63$; $p = 0.002$) and saccade length ($F(3,116) = 25.12$; $p < 0.001$). In Figure 6.6, one can see that DA and MA build a cluster due to their similar design. Furthermore, GA and WA are located in separated areas. The post-hoc analysis of the fixation duration reveals that there are differences between GA and all the other methods ($p < 0.05$). For the saccade length, all pairwise combinations show significant differences ($p < 0.007$), except between MA-DA. This confirms hypothesis H3, since MA, DA, and GA in fact lead to higher average saccade lengths.

With the AOIs for the outer region (*outside*) and inner region (*inside*), it is possible to analyze visual solution strategies; see Figure 6.4 for illustrations of the AOIs. The target of the analysis is to show that there are two phases of visual search for GA, DA, and MA: in the first phase, subjects focus on the annotation area *outside*, and switch then to *inside* in order to find the required label in the second phase. To extract these two phases, the mean timestamps of the fixations were used, separately for each AOI. Figure 6.7a shows that the two phases are clearly separated: fixations in *outside* happen in average much earlier than fixations in *inside*. For all annotation methods , except for WA, participants start searching in the outer regions because the mean timestamp of *outside* has a significantly lower value. For WA, there is no first phase and therefore no *outside*. This confirms the first part of hypothesis H4. The second part of H4 is analyzed by utilizing the average fixation durations, separately for both AOIs. Figure 6.7b shows the statistics plots. The impact of the two phases can be recognized. During the second phase, the average fixation duration is higher, since this corresponds to the search of the actual label by inspecting labels in a target area. For the first phase, GA shows higher fixation duration in comparison to MA and DA. For GA, participants not only had to find the reference to the grid cell where the label is located but also had to memorize the grid coordinates. Therefore, they had to focus more. For MA and DA, this is not the case because participants quickly identify the rough position and start the transition into the second phase, searching for the actual label. The significance test for *outside* indicates differences ($\chi^2(2) = 19.67; p < 0.001$). The



(a) Average timestamp (b) Average fixation duration (c) Average direction change

Figure 6.7 — Diagrams used to identify different subject behavior. (a) There are two phases during the solution of the task. (b) The fixation behavior is also different between the phases. The average direction change of saccades along scanpaths for all four methods is shown in (c).

post-hoc test shows that there are only differences between GA-MA ($p < 0.001$) and GA-DA ($p < 0.001$). GA achieved an average fixation duration of 406.17 ms, MA 311.92 ms, and DA 303.81 ms. For *inside*, there are also significant results ($\chi^2(3) = 25.89; p < 0.001$). In particular, there are differences between: GA-WA, MA-WA, and DA-WA with $p < 0.001$. GA achieved an average fixation duration of 451.76 ms for *inside*, MA 485.58 ms, DA 476.95 ms, and WA 384.08 ms. Based on these outcomes, the second part of hypothesis H4 can be confirmed as well.

Hypothesis H5 is analyzed concerning the average direction change along scanpaths at each fixation in the second phase. It is assumed that searching along a line results in smaller angular differences than searching in a rectangular area. To handle inversion of search direction (180° turns), the smaller angle of the change in direction is used.

The statistical tests indicate significant results concerning the annotation type ($F(3,116) = 34.84; p < 0.001$) and the post-hoc tests report differences for the following pairs: DA-WA, DA-GA, DA-MA, and GA-WA with $p < 0.001$. The average change in direction was 41.74° (GA), 39.34° (MA), 31.39° (DA), and 36.82° (WA); see Figure 6.7c. For DA, participants were changing the angle of direction less strongly, resulting in a more directed eye movement compared to the other methods. This indicates that the design of DA has an impact on the visual search behavior of the subjects, and therefore hypothesis H5 can be confirmed.

6.5.3 Qualitative and Subjective Evaluation

For the qualitative and subjective evaluation, participants were asked to fill out a questionnaire with the following questions regarding the different annotation methods:

1. Have you used a search strategy? If yes, please state it.
2. Have you used the annotations for orientation purposes?
3. Have you searched the annotations for a label in alphabetical order? (only for MA, GA, and DA)
4. Have you used the distance line marker for orientation purposes? (only for DA)
5. Which method allowed you to perform the visual search task most quickly?
6. Which method allowed you to identify the grid cell that contains the target label most quickly: MA or GA?

For questions 2 and 4, a Likert scale was used ranging from 1 (never) to 4 (always). For questions 1 and 3, *yes* and *no* answers were counted. For questions 5 and 6, the votes were counted for each map annotation method. In the following, the answers will be presented concerning the four methods.

Within-image annotation (WA): 19 participants stated that they used a search strategy. The most common strategies were horizontal or vertical scanning, or starting in the middle of the screen and progressing outward in a spiral.

Grid reference annotation (GA): 25 participants used a search strategy: They started at the annotations to search for the label and the associated grid coordinates and then moved to the grid cell. This correlates with the average answer for question 2, which is 3.7 on the Likert scale. Additionally, 23 participants were looking for the target label in the legend in alphabetical order.

Directional annotation (DA): Here, 26 subjects used a search strategy. First, they were searching for the directional annotation at the border (23 in alphabetical order), and next, they followed the arrow. An average Likert value of 3.73 was achieved for question 2, and a Likert value of 2.8 for question 4 (use of the distance line marker).

Miniature annotation (MA): A common search strategy was used by 25 subjects. Similar to DA, they first searched in the annotations (average Likert value of

3.8) for the correct label, and then jumped to the associated grid cell. 23 subjects stated that they benefited from the alphabetical order.

General questions: Questions 5 and 6 were dedicated to summarize the impression of the subjects about the map annotation methods. 21 participants stated that they could perform the search task fastest while using MA, followed by GA (7 votes) and DA (2 votes). Question 6 was supposed to compare MA and GA since they are both grid-based. Here, the participants favored MA with 23 votes.

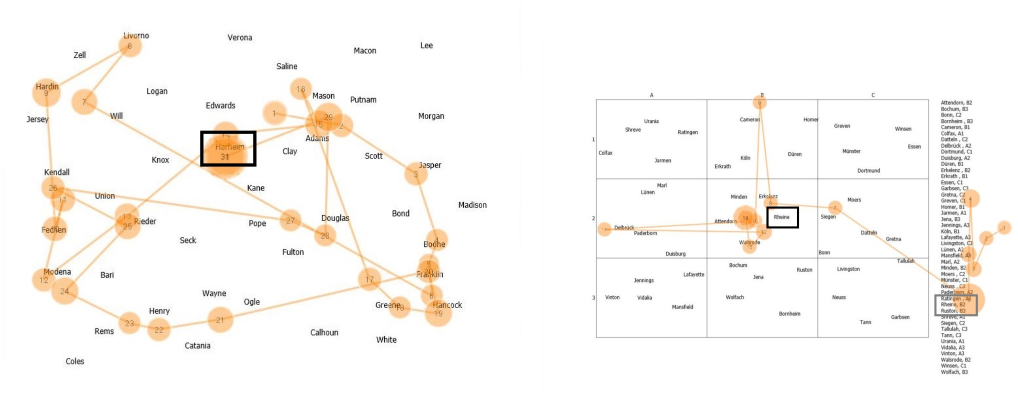
6.6 Discussion

Based on the results of the statistical evaluation in the previous section, several hypotheses from Section 6.4.1 could be confirmed.

It was verified (H1) that there are significant differences between WA and the other three map annotation methods. Participants were 23–40% faster in locating a specific label using the annotation methods with visual search support.

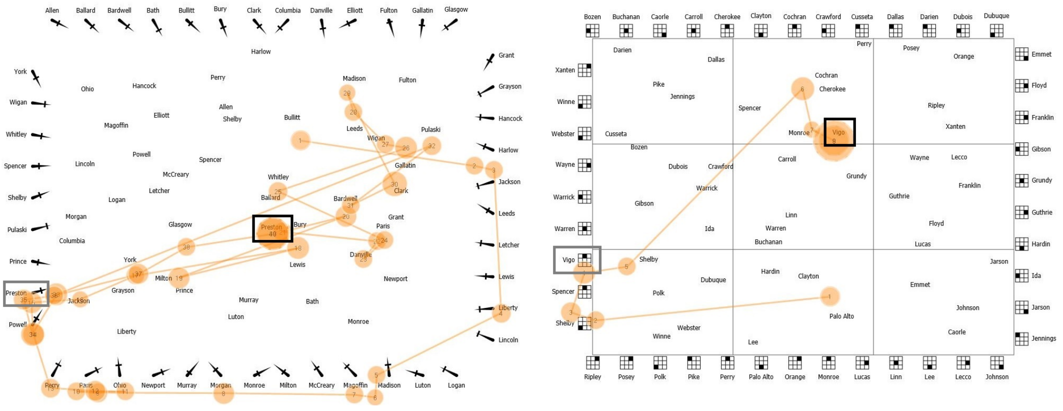
Hypothesis H2 was not fully accepted. The post-hoc analysis of completion times allowed ranking MA better than GA and DA. Based on the theoretical considerations from Section 6.3, the following potential outcomes could have been expected: On the one hand, DA could have performed better than MA, since DA provides directional and distance information, and therefore the search area should be more focused compared to the size of a grid cell. On the other hand, the performance of MA and GA could have been better than that of DA. Here, subjects might be more familiar with grid-based approaches, as they are used commonly in maps and atlases. Perceptual issues could also be a reason for the achieved performance of DA. It is well known that direction and distance estimation, especially for long distances, depend on many factors, e.g., arrow length or the number of distractor elements in the target area [181, 182, 183, 184]. For example, an arrow points to a specific point in a cluster. If the arrow is drawn with the full length of the target, it can be identified accurately. Shortening the arrow would introduce an error for the perceived direction in which the arrow is pointing. As a result, one of the distractors might be perceived as the target.

Concerning the eye-tracking evaluation, hypothesis H3 could be accepted. This is supported by the statistical tests and visible in Figure 6.6. Furthermore, it was shown that there are two phases during the solution of the task when annotations were provided. The first phase corresponds to the lookup of the target label *outside* of the map, while during the second phase, the participants were searching *inside* the map (see Figure 6.3). Since there is a separation



(a) Within-image annotation (WA)

(b) Grid reference annotation (GA)



(c) Directional annotation (DA)

(d) Miniature annotation (MA)

Figure 6.8 — Scanpaths of different participant behaviors for all four annotation methods tested in the study. Random search is shown in (a). In (b), there is first a lookup in the list followed by identifying the grid cell and finally searching for the label. In (c), first the annotation was used to get information about the direction and distance of the target, followed by a search in the respective direction. The initial step of (c) is also present in (d). In the second part, a search in the target cell is performed. The rectangles indicate the target label.

between the two phases, additional behavior characteristics can be inspected based on the average fixation durations for both phases. During the first phase, GA exhibits a higher fixation duration compared to the other methods, which could indicate that subjects required more time for reading and memorizing the respective coordinates (e.g., C2). During the second phase, all participants exhibited the same behavior: They were looking for the target label at the rough position provided by the position hint. Therefore, the fixation durations look alike, since reading a label and comparing it with the target label should take for all annotation methods approximately the same amount of time.

The design of the annotation method influenced the behavior of the subjects

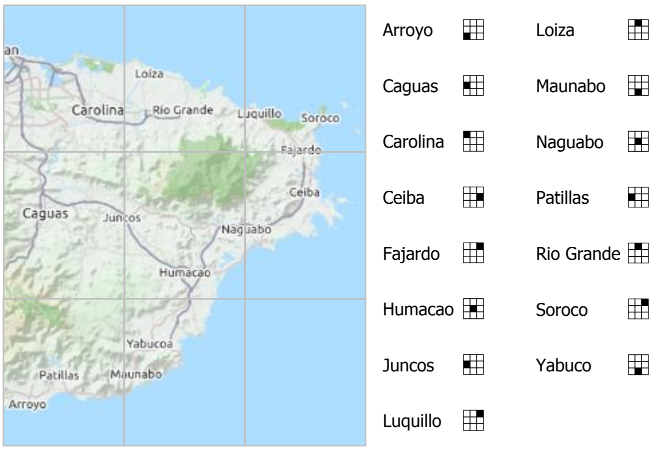


Figure 6.9 — Hybrid approach between GA and MA. Labels at the right side are combined with miniature representations.

(hypotheses H4 and H5). This is illustrated in Figure 6.1. Here, the attention is distributed differently for each annotation method. Looking at scanpaths also indicates different solution behaviors. Some example scanpaths are shown in Figure 6.8. Another behavioral aspect that was analyzed was if participants followed the arrow that points in the direction of the label for DA. It was shown that the relative direction change of the scanpaths was lowest for DA, indicating that subjects tend to move their eyes more linearly compared to the other methods. The evaluation in this chapter is using the evaluation approach of Chapter 5 as a basis, taking into account task performance and basic eye-tracking metrics. This sort of evaluation is here extended by using AOIs and their dwell times to verify that there are two distinct phases in the subjects' behavior: searching for annotation outside of the map and inside of the map. Participants do not constantly switch their gaze between outside and inside. Furthermore, a concept was introduced to test whether participants followed the directional hint of DA in a straight line. Thereby, the portfolio of evaluation methods, represented by the *statistical evaluation* block of the workflow in Figure II was extended. Here, the used evaluation methods revealed general behavioral patterns without using the need of visualization frameworks as described in Section 4.3 or 4.4.

The main drawback of miniature and directional annotations concerns the available space for placing the annotations at the border of the map. With an increasing number of labels, the size of the annotations at the border must be scaled down to avoid overlapping labels, leading to symbols that are hard to recognize. Although grid reference annotation also needs more space to show an increasing number of labels, the scalability issue is less pronounced because the annotation region is independent of the size of the map.

To increase the number of labels around the map, one could think of adding layers of annotations, instead of a single one as used in this study. The label has a huge impact on the space used by an annotation. Longer labels would require more space and, therefore, fewer annotations can be placed horizontally. Shortening their lengths, e.g., by using the shortest prefixes, would allow increasing the number of labels. However, there is still a limit on how many layers and labels can be placed around the map. For miniature annotations, a hybrid approach can overcome this problem completely. Here, the annotations can be displayed separately on additional pages similar to GA. This is illustrated in Figure 6.9. A hybrid approach was not tested in this study since the goal of this study was to isolate the effects of the different visual representations of positional information in the developed annotation methods.

As for most laboratory user studies, the design space and range of independent factors were limited due to constraints for the total procedure time. First, a constant number of labels was used (45 labels within each stimulus). Assuming that the lookup of a label is in general fast enough due to, e.g., alphabetical ordering of the labels. The effect of different label numbers was not tested in this study. However, the number of used labels was sufficient to test the ability of the annotations to support visual search in maps. Second, only one grid resolution (3×3) was tested for both MA and GA.

Nevertheless, baseline measurements were obtained through this study for the comparison with future study results with a subset of annotation methods and an extended parameter space (number of labels, grid size, etc.). Another interesting factor that could influence the results is the use of realistic instead of artificial maps since details like roads, contour lines, or other features could complicate or support visual search.

Although significant results were obtained from this study, the differences in task solution times are only within a couple of seconds. Based on the tested parameter space, an infrequent use would not benefit from annotations in practice. Furthermore, a potential drawback of MA and DA compared to GA is the loss of the ability to quickly communicate positional information verbally.

As a general guideline, it is conjectured that miniature annotations are a suitable concept for supporting visual search. It seems to be easier to follow a miniature representation of the domain compared to textual coordinates as in the case of GA. Furthermore, the search in the target area seems to be faster compared to DA. Potential applications that benefit from such annotations might not be restricted to maps and related scenarios. Possible use cases are discussed in the next section.

CHAPTER 7

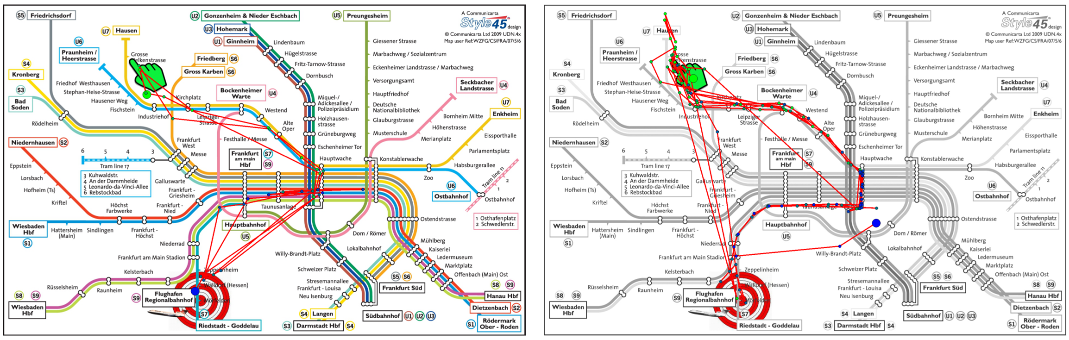
Evaluation of Metro Maps

In this chapter[1], the data of a controlled laboratory experiment with eye tracking and 40 participants is evaluated. The task of the experiment was to find a route from a highlighted start station to a highlighted destination station in metro maps. In total 24 different metro maps of big cities from around the world were used. To achieve similar characteristics among the map stimuli and to easily compare the recorded eye-movement data, all maps exhibited the same map layout style, as provided by the company *Communicarta Ltd.* (http://www.communicarta.com). Map complexity, color coding, and task difficulty (number of minimally needed interchange points) served as independent variables in the study.

The recorded study data was statistically evaluated concerning the measured response times, error rates, fixation durations, and saccades. Furthermore, the eye movements were annotated and statistically evaluated. The study shows that color plays a crucial role in improving task performance. However, color coding had no impact on the choice of reading strategy: similar patterns were found and quantified for color and gray-scale metro maps. Moreover, several reading strategies were identified such as a direct approach reading from start to

---

[1] **Parts of this chapter have been published in:**
R. Netzel, B. Ohlhausen, K. Kurzhals, R. Woods, M. Burch, and D. Weiskopf. User performance and reading strategies for metro maps: an eye tracking study. *Spatial Cognition & Computation*, 17(1-2):39–64, 2017 [8], © 2017 Taylor&Francis.



(a) (b)

Figure 7.1 — Metro maps for the city of Frankfurt with color (left) and gray-scale encoding (right). The eye tracking scanpaths are shown for two different participants, one for each color encoding. The big green dot indicates the start of the scanpath and the big blue one the end. The color for all fixations in between (smaller circles) is interpolated between green and blue to encode temporal information. The scanpath in the left image has a bimodality coefficient of $bc = 0.59$, and for the scanpath in the right image it is $bc = 0.64$.

destination and vice versa, and multiple reading cycles mostly used to confirm the result. An example for this is shown in Figure 7.1.

Metro maps, typically placed as a poster on station walls, are seen by millions of passengers daily where the goal of these maps is to provide a rapid overview of the public transport system. Not only is that overview important, so too is the relationship between the individual lines, stations, and interchange points. Schematic maps are designed to represent the transportation system by preserving the topological aspects. The distortion of areas where many stations are located close together, such as in a city center, is allowable and in many cases the direct effect of the topology-oriented map design.

The visualization of maps and routings has a long tradition. One of the first and oldest schematic maps is the *Tabula Peutingeriana*, which is inspired by the Roman world map that originated in the 4th century. Growing cities and the appearance of railways brought about an increasing need for public transport maps and 1874 one of the first maps that discarded almost all surface topography appeared [185]. The maps were further improved by distorting the actual geographic distances. In 1932, Henry Charles Beck introduced a ground-breaking design, exemplified by the London Underground schematic map, becoming the pioneer style for metro maps, of which many, to this day, contain his original ideas.

Many factors play crucial roles in effective metro map design, such as their color coding, display of the stations, labeling [186], and last but not least additional information like places where sightseeing attractions are located and how they can be reached. Despite many established design guidelines, there is no universally accepted standardization for metro map design.

To adapt metro maps to the needs of travelers and to improve the design of metro maps, designers and researchers are interested in how people read maps and how they solve given tasks, i.e., how they analyze the graphical map visually. In this context, eye tracking provides an excellent means for observing and measuring how maps are read.

7.1 Related Work

The term metro map is used to describe schematic maps of the kind epitomized by Henry Charles Beck [187]. Commentators and historians debate about who the real *father* of the metro map is. Some suggest that Beck was himself influenced by George Dow, who designed, among other works, a carriage diagram for the London & North Eastern Railway company (LNER). In Andrew Dow's book '*Telling the Passenger Where to Get Off*' [188], there is a fascinating look at the evolution of diagrammatic railway maps and the work of someone who, although nowhere near as famous as Beck, made some pioneering designs preceding Beck.

Although people have adapted to use the term metro map, maps could be referred to as transport maps, since often so-called metro maps are not only pure metro, they are a combination of a variety of transport modes. Having said that, some commentators and aficionados would argue that they are not even maps at all and should be referred to as diagrams [189]. Indeed, the London Underground map has been known, at different times during its history, to refer to itself exactly as a diagram of lines. In any case, it no longer portrays purely a metro system either now. The most recent editions include suburban surface rail (the so-called London Overground), light rail (Docklands), and even a cable car (so-called Emirates Air Line) across the Thames river.

For a detailed analysis of this user study, it is not sufficient to only measure traditional task performance. Therefore, eye-tracking information is collected about the reading behavior.

In general, eye tracking has become increasingly important for research in spatial and geographic information science. The application of eye tracking includes the evaluation of user attention [190], the influence on the readability of maps [191] as well as distinguishing ambient and focal attention [192]. Eye tracking can be used as a basis for a support system [193] and also facilitate the evaluation of spatiotemporal information change [194]. Furthermore, it is applied to study the visual interaction strategies of users while working with highly interactive interfaces [195]. However, none of these papers deals with the issue of how someone is reading metro maps.

A typical task related to metro maps can be described as a typical path search task. Therefore, there is some relationship to previous empirical studies in information visualization that investigate path searching in visual stimuli. In particular, node-link diagrams of graphs (networks) and trees (hierarchies) are loosely related to the visualization by metro maps. Huang et al. [196] discovered a geodesic-path tendency for a similar task in graph visualizations.

In Chapter 5 of this thesis, the influence of different trajectory visualizations on search tasks was investigated. To this point, the average saccade length, fixation duration, and task performance were statistically evaluated to describe the differences between the visualizations. More related to the topic of maps, an evaluation of a study related to finding labels using different methods of map annotations was presented in Chapter 6. Here, additionally, to the evaluation of task performance, AOI-based methods were used to infer user behavior.

Burch et al. [149] evaluated the differences between different node-link tree layouts with an eye-tracking study to explain diverging completion times between the layouts. The authors measured response times and error rates as standard dependent variables, but they also recorded eye movements that were later analyzed for visual task solution strategies [197]. Again, none of these studies considered metro maps. Although loosely related, metro maps come with many differences in the visual design and the exact tasks of these compared to other visualizations. For maps in general, design aspects like font size and label or symbol placement are important to make the map more appealing to the reader. Whereas a graph visualization aims to highlight inherent, important structures.

The above user studies mainly focused on the evaluation of standard metrics for their specific research question. If strategies were investigated, this was usually performed qualitatively.

Nevertheless, there are preliminary studies [198, 199] in this field. In this work, metro maps from different web sources were presented to 8 participants, and a qualitative visual analysis of gaze plots was performed to identify different strategies for the route finding task.

Although that work provided valuable first insights into how people read metro maps, neither quantitative nor robust evaluation of the extracted findings could be performed.

Considering the evaluation of visual search strategies, many approaches are based on string comparisons. These strings usually describe a participant's scanpath by sequential visits of areas of interest on a visual stimulus where each area is represented by a label. For the evaluation, statistical approaches to identify the longest common subsequence (e.g., Yesilada et al. [200]) or sequential pattern mining (e.g., Hejmady et al. [201]) can be applied. For this approach, only a small number of areas is defined, to keep the number of possible labels low. However, in the evaluation of metro maps, the definition of areas of interest would require annotating all stations and lines individually. This approach would result in a large number of areas, which usually leads to very divergent string representations. Also, due to the visual span of a participant, the dense parallel lines cause uncertainty about which of them was investigated. This uncertainty impairs the evaluation of strategies.

7.2 Experiment Design

This section describes the design, hypotheses, tasks, stimuli, and further details of this user study.

7.2.1 Hypotheses and Research Questions

The overarching goal was to understand reading strategies for metro maps. Individual hypotheses that are tractable by a controlled eye-tracking study were established. In particular, the following dependent variables were analyzed: response times, error rates, and eye movements. Since eye movements per se cannot be treated as a dependent variable, particular characteristics of them were used (fixation duration and saccade length) to measure the performance on reading maps.

Map complexity level, task difficulty level, and color encoding were considered as independent variables. By varying these independent variables, different reading behaviors are elicited to investigate reading strategies.

The associated hypotheses are split into three groups, corresponding to the type of dependent variable used. Hypotheses 1 to 3 are related to task performance:

H1: Participants are faster and more accurate in color metro maps than in gray-scale maps, because individual lines can be distinguished more easily when shown in color.

H2: With increasing complexity, the response time will increase.

H3: If the map complexity is low, the difference regarding response times and error rates between colored and gray-scale maps is small (as opposed to the case of maps of high complexity), because the chance of misinterpretations will be smaller for low complexity.

The following hypotheses focus on eye-tracking metrics:

H4: In color-coded metro maps, the saccades will be longer, and the fixation durations will be shorter than in gray-scale maps. The colored lines will help the participants identify the desired line faster (i.e., with lower fixation duration) even if the participants will cover a long distance with eye movements (i.e., longer saccade length).

H5: With increasing map complexity, the saccades will be shorter, and the fixation durations will be longer. For high map complexity, a participant will need more time to identify a line (i.e., longer fixation duration), due to a more densely filled map. For the same reason, the distance a participant will cover with one eye movement will be shorter (i.e., shorter saccade length), since the participants do not want to *lose* the line they are following.

Hypotheses 6 and seven are related to the visual strategies:

H6: There will be distinguishable reading strategies that can be separated clearly.

H7: There are three main visual task solution strategies: (1) find a route directly from a start to an end location, (2) go from start to end and backward verifying the result, (3) use multiple repetitions of going back and forth.

The hypotheses are also motivated by the suggestions of the cooperation partner *Communicarta Ltd*. A commercial aspect here would be, whether it is worth printing and selling gray-scale pocket maps instead of more expensive colored ones. However, this depends heavily on how good the performance with gray-scale maps is.

The extraction of reading strategies could also be of use to enhance the design of metro maps. If strategies show that participants have problems using a map and where these problems are located, it could be redesigned.

Table 7.1 — Abbreviations for the transfer groups and map complexities.

U0	No transfer needed
U1	At least one transfer needed
U2	At least two transfers needed
C1	Low map complexity
C2	Medium map complexity
C3	High map complexity

7.2.2 Study Design

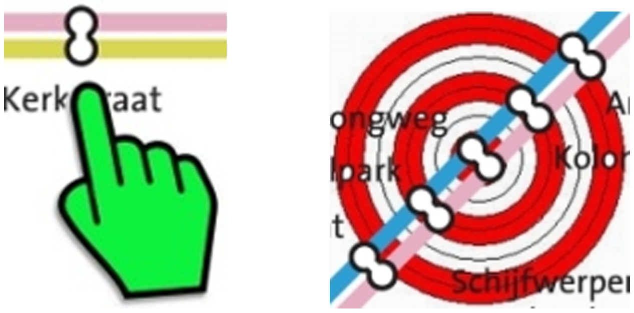
The study was designed as a mixed within-/between-subjects study design. The independent variables of interest are: complexity, transfer group, and color coding of the metro maps. The abbreviations for complexities and transfer groups are summarized in Table 7.1.

Complexity: We categorized 24 preselected metro maps of different cities from around the world into three different complexity classes: 8 small ones (C1), 8 medium-complex ones (C2), and 8 highly complex ones (C3), which were categorized by inspecting the number of lines, stations, and interchange points. The individual complexity values ranged in C1 from 71 to 121, in C2 from 122 to 150, and in C3 from 151 to 314. The maps are chosen in a way that the range of each group is sampled equally.

Transfer group: With this factor, task difficulty levels was controlled. The transfer group describes the minimum number of changes that are needed to get from a start to a destination station. Such station pairs were defined for each map. For each map complexity, this results in two maps where participants did not have to change lines at all (U0), three maps where they had to change at least once (U1), and three maps where they had to change at least twice (U2).

Color coding: Color is used in metro maps as a visual feature to support the participant in route finding tasks. We varied between color-coded maps and those only showing gray-scale.

As part of the design, participants were randomly mapped to one of two participant groups. Both worked with different metro map stimuli (to avoid learning effects). In each group, a within-subjects study design was established. The participants were supposed to solve the task for 48 stimuli that were divided into two blocks, each containing 24 stimuli. One block contained color-coded



(a) Start (b) Destination

Figure 7.2 — The start and destination markings. A green hand indicates the start station (a). A red dartboard shows the destination station (b).

maps and the other one a gray-scale version of the same maps. For each stimulus, a unique set of start and destination stations was assigned. The first group solved the tasks in the described manner. For the second group, the order of color-coded vs. gray-scale stimuli was swapped to compensate for order effects; here, the start and destination stayed the same for each stimulus. The stimuli were presented in a blocked manner and not in a random order to avoid that participants were confronted with a frequently changing color encoding, which could affect the performance.

Subjects were asked to find a route from a start station to a target station. There were no restrictions given to the participants concerning their search behavior by asking, e.g., for the shortest route. The task here was strongly motivated by real-world scenarios, e.g., being in an unknown city and finding a route from a current location to a destination. Here, no one would tell a person to find the shortest path or the route with a minimal number of transfers. In this way, the results are more natural instead of forced by applying mentally a predefined metric, while solving the task. The start and target stations were highlighted using a green hand and a red dartboard (Figure 7.2) to avoid that subjects scan the map to search for them. While searching for a route between start and target stations, participants had to inform the experimenter about the number of line changes.

7.2.3 Stimuli

The high-quality standardized metro maps used in this study were provided by the cooperation partner *Communicarta Ltd.*, a well-established company with more than 25 years of experience in creating, designing, and researching public transport information and metro maps. The maps were scaled down to achieve a height of 1200 pixels—the maximum vertical resolution of the display. The horizontal resolution was on average 1605 pixels (minimum of 871 pixels and maximum of 1894 pixels).

All maps used in this study share the same style to make them comparable and to avoid a possible influence of the layout on the reading behavior. The style has several specific characteristics that can be described as follows:

Lines: Typically, they are shown as color-coded lines. Each line is attached to a unique number or letter that is placed at both end stations shown in Figure 7.3a.

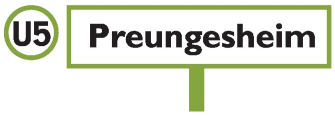
Stations: Figure 7.3b depicts different examples of stations. They are indicated by a small arrow head connected to the corresponding line or as white filled circles surrounded by a black circle (in case of an interchange point). The names of the stations are placed next to the arrow heads or circles always written in a horizontal orientation. If the mode of transport only runs in one direction, this is indicated by a white arrow pointing into the direction.

Interchange points: A certain number of lines are crossing an interchange point. This is indicated by a white filled circle with a black surrounding circle placed on top of the line at the corresponding position of the station. At an interchange point, all lines can be traveled which are attached by white circles and a connecting white line. If this line is only a black one, travelers have to walk about 20 to 75 meters. At an interchange point where the mode of transport only stops in one direction, there is a white arrow on top of the corresponding line. Interchange points are illustrated in Figure 7.3d.

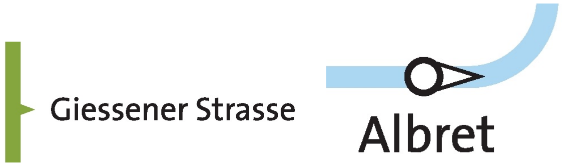
End stations: The names of the end stations are indicated by a rectangle in the same color of the line. If there is an interchange point, the rectangle is black or divided into a black and colored rectangle. Examples are shown in Figure 7.3c.

Each color-coded map was transformed into a corresponding gray-scale map using Adobe Photoshop since color-coded and gray-scale metro maps were compared in this study. This is exemplified in Figure 7.1 for two maps of the same city but with different color encodings.

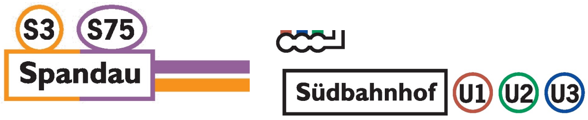
Metro maps from major cities all over the world were used to perform a sort of real-life study using real maps. Therefore, artificially created maps were not



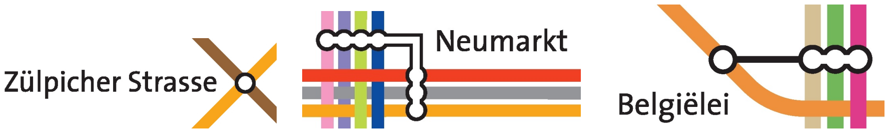
(a) Lines



(b) Stations



(c) End stations



(d) Interchange points

Figure 7.3 — Characteristics of the layout style for the generation of the metro maps used in this study.

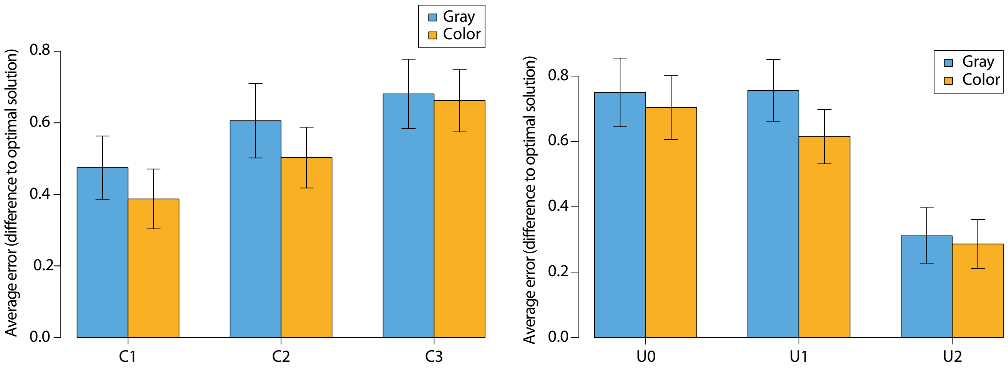
used. All maps were independently designed, none are the official version used in the public transport of the cities. Even if participants had been to the city and used the transport system, it is unlikely they would have used the maps utilized in this study.

7.3 Results

In addition to the methodology described in Section 2.3 for a statistical analysis, a clustering of reading strategies was applied. This will be described in more detail in the Section 7.3.3 after task performance and eye-tracking data are evaluated.

7.3.1 Task Performance

Task performance was analyzed concerning correctness and completion time.



(a) Complexity (b) Transfer group

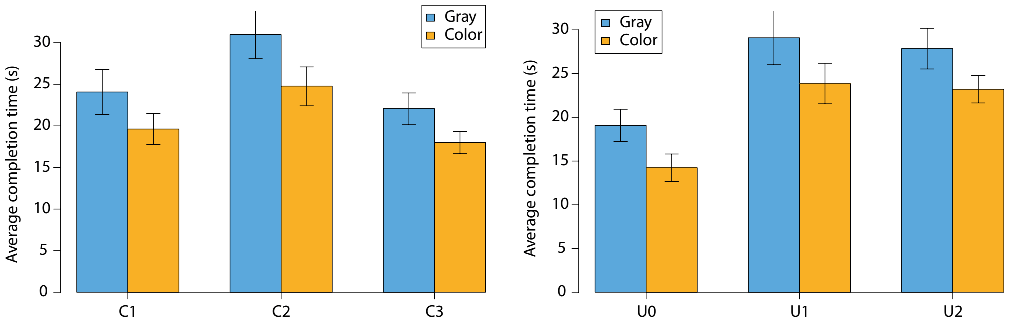
Figure 7.4 — The average difference between the minimal needed number of transfers and the number of transfers users needed to solve a task for the color encoding. (a) shows the results for the complexity and (b) for the transfer group. Error bars show the standard error of the means (SEM).

Correctness: Due to ambiguities of correct solutions for every task, correctness cannot be calculated and interpreted conventionally. Instead, the minimum number of transfers necessary per task to solve it was determined and the differences to the participants' answers were used as an indirect measure of the error.

With an average in difference of 0.52, colored metro maps achieved better correctness (smaller error) than gray-scale maps with an average difference of 0.59. Here, the statistical test indicates no significant result between these two groups. However, having a closer look at the different transfer groups (task difficulty) and complexities, the tests indicate significance for both: transfer groups ($\chi^2(2) = 45.09$; $p < 0.001$) and complexities ($\chi^2(2) = 27.33$; $p < 0.001$). The post-hoc analysis shows that there are differences between transfer groups U0-U2 and also U1-U2 with $p < 0.001$. Regarding complexity, there is a significance between C1-C3 with $p < 0.001$.

Figures 7.4a and 7.4b support this outcome. Considering the complexity (Figure 7.4a), the error behaves as expected: with increasing complexity the error increases. Using color-encoded maps leads to smaller errors than using gray-scale maps for each complexity.

For the transfer groups (Figure 7.4b), it is the opposite: increasing the number of minimally needed transfers leads to smaller errors. This indicates that users tend to choose longer ways or that they are simply not able to identify routes with fewer transfers.



(a) Complexity (b) Transfer group

Figure 7.5 — The average completion time for each complexity (a) and transfer group (b) with respect to the color encoding. Error bars show the standard error of the means (SEM).

Completion Time: Color-encoded maps achieved here on average a completion time of 20.80 s, which is better than the average time of 25.71 s for gray-scale maps. According to the statistical test, this is a significant result ($F(1{,}39) = 36.23$; $p < 0.001$). By examining the different complexities and transfer groups more closely, statistical tests indicate a significance for both: map complexity ($F(2{,}78) = 95.21$; $p < 0.001$) and transfer groups ($F(2{,}78) = 157.20$; $p < 0.001$). The corresponding bar charts are depicted in Figures 7.5a and 7.5b. These charts indicate that there could be statistically significant differences, which was confirmed with a post-hoc analysis: there are significant differences between C1-C2 ($p = 0.004$), C2-C3 ($p < 0.001$), U0-U1, and U0-U2 ($p < 0.001$). For the transfer groups, the completion time behaves as expected: the more transfers are needed, the longer the completion time. Comparing the error (Figure 7.4b) and the time (Figure 7.5b), a correlation was identified: the answer will be more accurate if the user takes longer to solve the task. In other words, the longer a user is looking for a route, the lower the number of transfers will be. The Pearson correlation test indicates, with a value of 0.14, a slightly positive linear relationship. The results show that on average the response time for color-encoded maps is lower compared to that of gray-scale maps. Considering the map complexity and the task difficulty, differences were verified, which are illustrated in Figures 7.5a and 7.5b. Here, color encoding always achieves a lower response time. Regarding the solution error, defined as the absolute difference of transfers between the correct solution and the user's answer, color-encoded maps achieved better results. This is depicted in Figures 7.4a and 7.4b. Supported by this statistical analysis hypothesis H1 can be confirmed.

An interesting aspect here is that the completion time first rises and then falls again, regarding the increasing complexity (Figure 7.5a). At the same time the error rate is constantly increasing (Figure 7.4a). Having a closer look at the task difficulty, is it apparent that the response time increases with the difficulty (Figure 7.5b), and at the same time, the error decreases (Figure 7.4b). Overall, this means that the users tend to select routes that are inefficient, but that can be identified with ease, and that they tend to select routes with more transfers in between than necessary. Hence, hypothesis H2 could not be fully confirmed.

The results show that H3 cannot be confirmed. The opposite effect occurs: for low complexities, the difference in response time and the error rates is larger or equal compared to higher complexities (see Figures 7.4a and 7.4b).

7.3.2 Eye Tracking

Now a statistical evaluation of data obtained by eye tracking is presented based on common eye-tracking metrics, namely the average saccade length and the average fixation duration.

Using such kinds of metrics, the performance and influence of different key aspects can be interpreted in more detail as previously described in Chapter 5 and 6. A possible interpretation of those is described in Section 2.2.2.

Nevertheless, perceptual and cognitive processes are by nature highly complex, and therefore the interpretation of eye-tracking metrics should be used with caution. However, these metrics add a useful view on this user study.

Color Coding: Figure 7.6 shows that color-encoded maps achieved an average saccade length of 178.18 pixels and an average fixation duration of 347.28 ms, whereas the average values of gray-scale are 153.86 pixels and 373.68 ms. Statistical tests indicate a significance for both the average saccade length ($F(1{,}39) = 158.8$; $p < 0.001$) and fixation duration ($F(1{,}39) = 37.62$; $p < 0.001$). Therefore, color-encoded maps outperform gray-scale maps concerning the interpretation of desired saccade and fixation values. This is in accordance with the expectations: participants use more explorative eye movement (larger saccade length) and can identify relevant information—here the different lines—more easily while using color-encoded maps.

Map Complexity: The map complexity has a significant influence on the performance. The test reveals here significance for both average saccade length ($F(2{,}78) = 47.98$; $p < 0.001$) and fixation duration ($F(2{,}78) = 54.91$; $p < 0.001$). C1 achieved average values of 180.11 pixels and 338.94 ms, C2 values of 160.21 pixels and 363.62 ms, and C3 on average 157.78 pixels and 378.83 ms. The post-hoc analysis shows again that there are differences between the complexities for both the saccade length and fixation duration: namely C1-

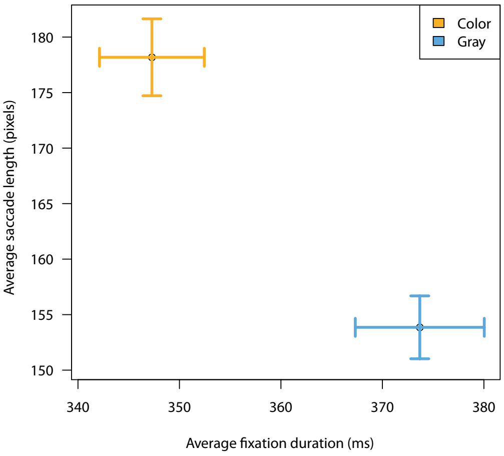
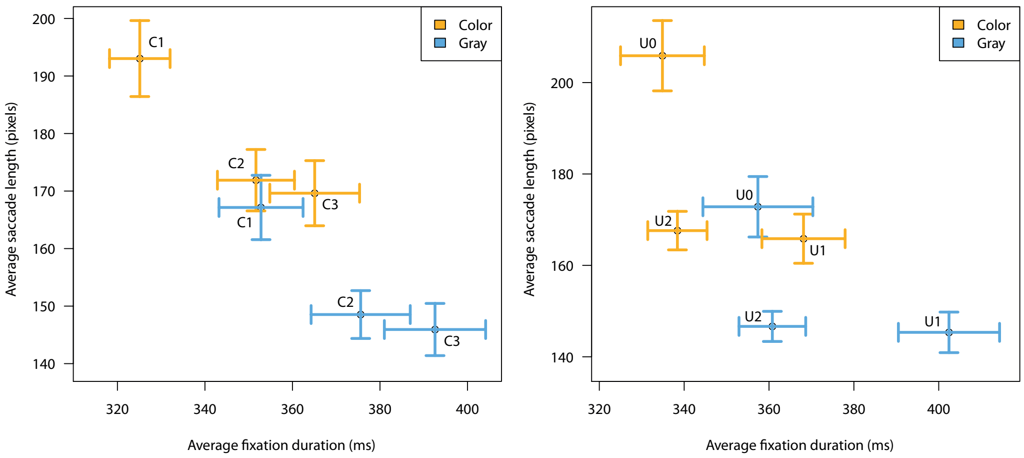


Figure 7.6 — Average saccade length (y-axis) and average fixation duration (x-axis) for the color encodings. Error bars show the standard error of the means (SEM) for the eye-tracking data.



(a) Complexity

(b) Transfer group

Figure 7.7 — Average saccade length (y-axis) and average fixation duration (x-axis) for the color encodings with respect to the complexity (a) and the transfer groups (b). Error bars show the standard error of the means (SEM) for the eye-tracking data.

C2 ($p < 0.001$) and C1-C3 ($p < 0.001$). Therefore, C1 achieved the best results, followed by C2 and C3. The average fixation durations and saccade lengths for the map complexities are shown in Figure 7.7a. Here, the map complexities are plotted separately for both color encodings. The effects are visible for both the color-encoded and gray-scale maps.

Transfer Groups: The statistical test for the transfer groups shows significant results for the average saccade length ($F(2{,}78) = 165.80$; $p < 0.001$) and fixation duration ($F(2{,}78) = 77.36$; $p < 0.001$). U0 achieved an average saccade length of 189.35 pixels and on average a fixation duration of 346.13 ms. U2 reached 155.59 pixels and 349.58 ms, and C1 had on average 157.14 pixels and 385.28 ms. The post-hoc pairwise comparisons show significance for saccade length between U0-U1 and U0-U2 with $p < 0.001$ and also for the fixation duration between U0-U1 and U1-U2 with $p < 0.001$. Since a large average saccade length and small fixation duration is considered as good, U0 achieved the best result followed by U2 and U1. The described facts are visible in Figure 7.7b. The different transfer groups are here separated into the different color encodings.

These results indicate that H4 be confirmed by inspecting Figure 7.6. Color-encoded and gray-scale maps are separated, and color-encoded maps outperform the gray-scale maps with a long average saccade length and a short fixation duration. A closer look at the complexities and the transfer groups reveals here for both a distinct pattern replicated in the lower right and top left area. The separation into the different complexities in Figure 7.7a shows that the replicated pattern results in three clusters: at the top left a color map cluster, bottom right a gray-scale map cluster, and a mixture in the central area. This is also the case for the different transfer groups in Figure 7.7b. At this point, H5 can be confirmed, since the average fixation duration is in fact increasing and the average saccade length is decreasing with rising complexity. This is valid for color-encoded and gray-scale maps.

7.3.3 Reading Strategies

For the analysis of the reading strategies, 39,404 fixations were annotated manually of one of the two groups of the user study. A label was assigned for each fixation that characterizes the action a participant was most likely performing. Subjects also include information obtained through their peripheral vision to determine their next action, but a subsequent annotation including possible information from peripheral vision is difficult without any further recorded information during the experiment. A think-aloud study where the participant comments all actions could be of help, but this would be a distraction while solving the task. Since the design of the study and the task requires a close tracing of paths, foveal information can be assumed to be the driving aspect here. Therefore, this study is analyzed based on the foveal information gathered by the eye-tracking device.

The scanpath-oriented method described in Section 4.2 of this thesis was used

Table 7.2 — Description of annotation labels.

J	Jump – the participant directed the visual attention to a different area, starting a new action there that did not follow-up from the previous fixation
NJ	Nothing/Jump – combination of J and N
JD	Jumping in Direction – the participant was trying to follow a line, but here fixated a position away from the line; the previous and subsequent fixations were on the line, though
JF	Jump/Follow – the participant was following a line, even after skipping a large distance
F	Follow – the participant was following a line
T	Transfer – the participant was changing the line
FTR	Follow/Transfer – combination of F and T
FTU	Follow/Turn – the participant was following a line and changed direction
N	Nothing – the participant was looking at an area that could not be associated with any object of the map
S	Start – the participant was looking at the start location
E	End – the participant was looking at the destination
FILA	Marks the start and the end of the eye-tracking trajectory

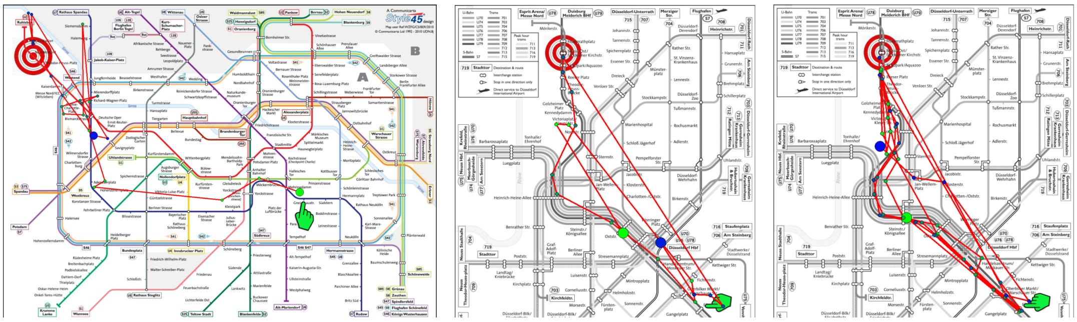
for the annotation of fixations. An automatic annotation of fixations that covers certain AOIs is not possible since the recording error of the eye-tracking device is of the same magnitude as the size of the AOIs. Therefore, it is possible that despite the fact that a participant is looking at a specific AOI, a fixation could be recorded outside of it, which would wrongly mean that a subject was not looking at the AOI. For these reasons, manual annotation was performed compensating for recording errors and utilized the sub-scanpath to assign a semantic label. Nevertheless, automatic annotation is possible to a certain degree, e.g., Kiefer et al. [202] proposed an approach that maps fixation sequences to a sequence of feature vectors of a geographic vector map.

Two previous and two subsequent fixations were taken into account to assign an appropriate label to the center fixation from a more global perspective instead of a local perspective by considering only the center fixation. This mimics to a certain degree the peripheral information that led to an action. This includes saccade information indirectly—through the previous and subsequent fixations. Allover, 12 labels were defined that cover four areas: path following, large jumps, line transfer, and start/end locations. A detailed description of the labels is given in Table 7.2. The semantics of the annotation often implies information about saccades; here, the saccade information is associated with the fixation that follows this saccade. As an example, the label *J* is associated with a fixation on an object of the map, following a *jump* (saccade) from a different area. Alternatively, the label *F* is used when a saccade led from a previous fixation on a line to the current fixation along the same line. For the annotation, the inaccuracy of the eye-tracking device and the size of the screen area covered by the fovea (approximately a visual angle of two degrees) was taken into account by considering lines that were in that uncertainty range. The annotation was performed by one person to ensure a constant quality of the annotation. The labels of the annotation were used to compute normalized transition matrices for each scanpath: 2D for two and 3D for three subsequent fixations. A trend was identified from the 2D matrices for the most commonly used transitions. The most frequent transition was from *path following* to *path following* (10.5%), followed by *path following* and *transferring* (8.3%) or the opposite (8.2%). Considering the jump labels, the most frequent pairs are the transitions of *looking at a blank space* and *jumping back to look at lines* (7.7%), followed by *jumping* and *jumping* (6.2%) and also from *jumping* to *path following* (5.6%). Note that only transition pairs are stated that appear with a frequency of more than 5%. To identify possible reading strategies, two different approaches were used. The first one is a clustering method and the second one utilizes an orthogonal projection of fixations of a scanpath onto a geodesic line between start and end location. Using the distance from the start location to the projected fixations results in a geodesic distance plot. The distribution of the distances is now used to calculate a bimodality coefficient, which was used for quantification.

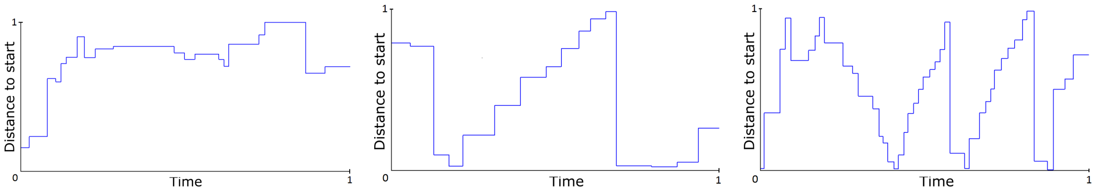
The projection was performed for every scanpath, which reduced the domain from 2D to 1D and made it possible to compare different maps based on a shared property: the distance from the start location.

Clustering: To automatically process scanpaths to extract possible strategy classes a clustering approach was used. As input for the clustering, 3D transition matrices were calculated for each scanpath and interpreted as feature vectors of length 12^3. The transition matrix contains the information about how often a transition from one label to another occurred based on subsequent fixations of the scanpath. This means if two elements will be assigned to the same cluster they should exhibit a similar transition fingerprint supposing they are close to each other in high-dimensional space. One encountered problem was that the completion time (respectively the scanpath length) influenced the clustering. The result was that long scanpaths were closer to the cluster center than shorter ones. Nevertheless, similar behavior patterns could be found qualitatively.

To reduce the influence of the scanpath length, the results of the clustering were



(a) Scanpath cluster 0 (b) Scanpath cluster 1 (c) Scanpath cluster 2

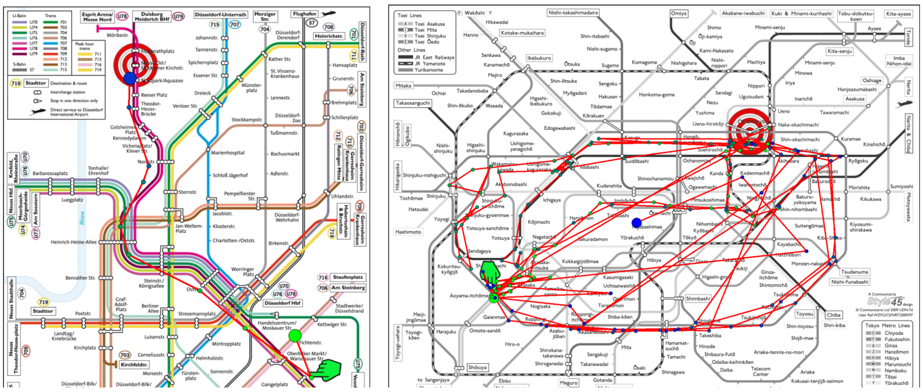


(d) Plot of (a) (bc = 0.62) (e) Plot of (b) (bc = 0.52) (f) Plot of (c) (bc = 0.51)

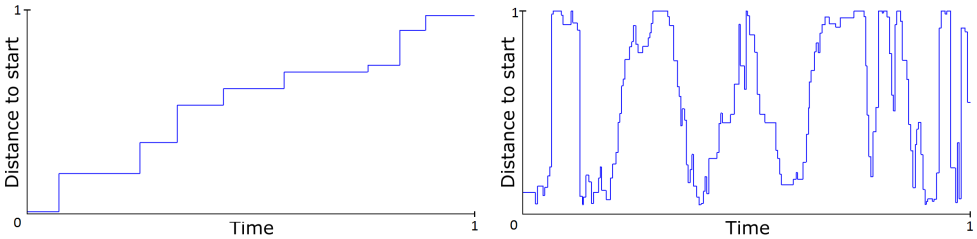
Figure 7.8 — Three example scanpaths are depicted in (a), (b), and (c); they are close to their individual cluster centers of clustering group P1. The big green dot indicates the start of the scanpath and the big blue one the end. The color for all fixations in between (smaller circles) is interpolated between green and blue to encode temporal information. In (d), (e), and (f), the associated projected geodesic distance plots are shown together with their bimodality coefficient *bc*. The x-axis corresponds to time and the y-axis to the distance to the start location. Time and distance values are normalized to a range of [0,1]. Horizontal blue lines indicate fixations and vertical lines indicate saccades.

improved by limiting the deviation of the distance to centroids of the clusters to less than 0.05. This was achieved by first calculating the average scanpath length of each stimulus and splitting the stimuli into four parts from a short (P1) to a long average length (P4). Then, a clustering was applied to each group. Performing the clustering on all the scanpaths led to a standard deviation (sd) of 0.15 and a mean distance (md) of 0.25. For P1 the values are $sd = 0.05$ and $md = 0.23$, P2 $sd = 0.04$ and $md = 0.18$, P3 $sd = 0.03$ and $md = 0.15$, and for P4 $sd = 0.03$ and $md = 0.13$.

The x-means clustering algorithm resulted in a cluster number of three for each group. Some examples of the predominant strategies in P1 are shown in Figure 7.8. The elements in the other groups exhibit the same tendency but with longer scanpaths and more complex strategies. In an extreme case, this results in several cycles of going back and forth between start and end location. This is shown in Figure 7.9 together with the opposite extreme case where a route was found directly.



(a) Short scanpath (b) Long scanpath



(c) Distance plot for (a) (bc = 0.37) (d) Distance plot for (b) (bc = 0.65)

Figure 7.9 — Examples of short (a) and long (b) scanpaths. The big green dot indicates the start of the scanpath and the big blue one the end. The color for all fixations in between (smaller circles) is interpolated between green and blue to encode temporal information. In (c) and (d), the associated projected geodesic distance plots are shown together with their bimodality coefficient *bc*. The x-axis corresponds to time and the y-axis to the distance to the start location. Time and distance values are normalized to a range of [0,1]. Horizontal blue lines indicate fixations and vertical lines indicate saccades.

Statistical Inference: To obtain a characteristic value that describes the geodesic distance plot, the bimodality coefficient ($bc \in [0,1]$) was calculated. A value of 5/9 stands for a uniform or exponential distribution. An indicator for a bimodal or multimodal distribution is a value greater than 5/9. The maximum value of 1.0 is only reached by a Bernoulli distribution or if there are only two distinct values, e.g., the sum of two Dirac functions. For one scanpath, the calculation of the bimodality coefficient is performed as followed: The fixations of the scanpath are projected on the geodesic (straight) line from the start to the end location. Then the distance from the start location to each projected fixation is computed. This list of distances is used to calculate the coefficient.

The interesting question is now if this coefficient can be used to find significant differences regarding different aspects. Therefore, a statistical analysis was conducted for the complexity ($\chi^2(2) = 42.20$; $p < 0.001$), transfer groups ($\chi^2(2) = 23.40$; $p < 0.001$), and clustering groups ($\chi^2(3) = 54.81$; $p < 0.001$), which showed significant results. The test for the color encoding did not show any significance.

The post-hoc tests indicate differences between the subcategories. For the complexity, there are C1-C2 ($p = 0.008$), C1-C3 ($p < 0.001$), and C2-C3 ($p < 0.001$). The transfer groups differ in U0-U1 ($p = 0.001$), U0-U2 ($p = 0.037$), and U1-U2 ($p = 0.025$). The significant results for the clustering groups are P1-P2 ($p < 0.001$), P1-P3 ($p < 0.001$), P1-P4 ($p < 0.001$), P2-P4 ($p = 0.002$), and P3-P4 ($p < 0.001$). The corresponding boxplots are shown in Figure 7.10.

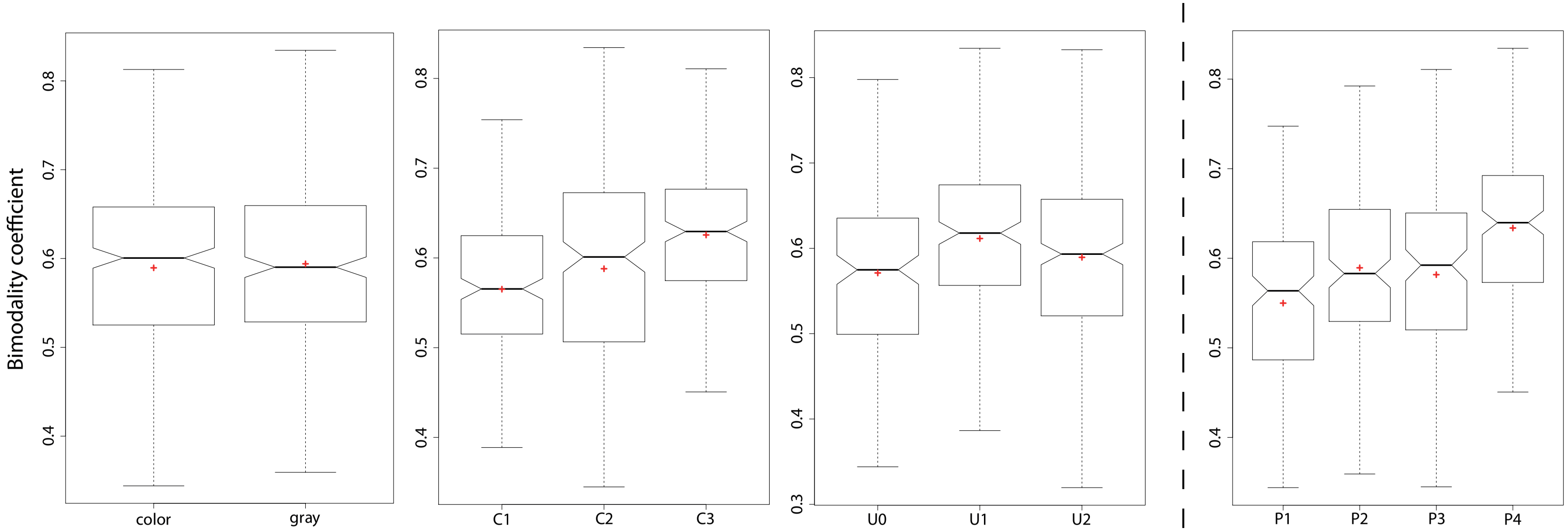
This means that the distributions that the bimodality coefficients represent change between the different classes within a test. Therefore, the bimodality coefficients can be used for quantification and comparison. As shown in Figures 7.8 and 7.9, the bimodality coefficients (*bc*) of the distance plots vary if the strategy is changing. Similar values point to the usage of a similar strategy, which is depicted in Figures 7.8e and 7.8f.

Based on these outcomes, hypothesis H6 could be confirmed. The initial evaluation by using a clustering upon 3D transition matrices has different clusters that were at first evaluated qualitatively. That indicated that there are different types of strategies that are clearly separated.

In the next step, quantification was performed by calculating the bimodality coefficient for each scanpath. The results show (Figures 7.8 and 7.9) that different values of the coefficient indicate different reading patterns. Also, a statistical analysis based on the complexity, transfer groups, and clustering groups showed that there are significant results (Figure 7.10). Qualitatively, the presence of three different solution strategies could be identified using geodesic distance plots. Figures 7.8e and 7.8f show a verification of the result by going back and forth one or multiple times. Figure 7.9c shows a direct approach without verification. The quantitative evaluation supports these findings since the bimodality coefficients are different and exhibit statistical significance. Therefore, hypothesis H7 can be confirmed.

7.4 Discussion

When using color-encoded metro maps, participants achieved better task performance concerning the response time and correctness, which confirms H1. H2 was confirmed partially, since the response time started to decrease, while the complexity was increasing. However, the error was increasing at the same time, which means that a route was found that is not optimal but that could be



(a) Color encoding (b) Complexity (c) Transfer group (d) Clustering group

Figure 7.10 — Boxplots of the four factors color coding, complexity, transfer group, and clustering group concerning the bimodal coefficient. The red cross marks the mean value while the black bar indicates the median. The notches at the sides of each box show the confidence intervals.

found fast. It could be possible that the longer alternative route was running through less densely populated areas and therefore stood out on the map.

H3 is the hypothesis that was rejected since the opposite effect occurred: the difference of response time and correctness between color-encoded and gray-scale maps drops with increasing complexity. This indicates that the use of color could at a certain complexity level no longer be useful. This is also supported by the evaluation of the eye-tracking results. Here, is can be observed that an increase of the complexity will lead to a reduced average saccade length and an increased fixation duration. This means that users tend to follow lines more closely and therefore need more time to process regions at which they are looking since they have to identify one line in a densely populated region. In accordance with the results, H4 and H5 can be confirmed. In this study, there were only used three complexity levels. Therefore it would be interesting to extend the research in order to find the threshold where two tested color encodings perform equally well.

The clustering applied to disjoint subsets of the 3D transition matrices of the scanpaths was used to find different high-level behavior clusters within each subset. This confirmed H6. Using the proposed bimodality coefficient in a further analysis was an approach toward quantification of reading strategies. Here, similar strategies achieved similar coefficients. The evaluation also showed that there is no significant difference between the strategies that are used in color-encoded and gray-scale maps. This is exemplified in Figure 7.1. Both scanpaths are within clustering group P3 and were assigned to cluster 0. The scanpath of the color image has a coefficient of 0.59 and the scanpath of the gray image 0.64. This result of the study is interesting: there is no difference in the reading strategy for color-encoded and gray-scale maps. One could have had hypothesized that color could support longer saccades with different characteristics of *jumping* through the image—but that could not be confirmed. The effect of independent factors on the choice of reading strategy could be explored in more depth in a follow-up study. While color/gray-scale did not have an effect, there might be thresholds for the complexity at which participants will change their strategy.

The bimodality coefficient allowed the interpretation of how the attention of users was distributed on geodesic lines. Therefore, different strategies could be identified. However, there are other coefficients that could be used to achieve a separation into different strategies as described in Section 2.2.2.

In contrast to the methods presented in Section 4.3 or 4.4, where groups of participants with similar behavior had to be found visually, the approach presented in this chapter provides an automatic grouping of similar scanpaths based on labeled fixations and, thereby, also of participants. Statistical tests were used to test for differences between the groups. This is an improvement compared to the evaluation in Chapter 6, where only general behavioral patterns could be shown. Furthermore, this corresponds to a hybrid approach between a data-driven analysis and the methods associated with the *statistical evaluation* block in Figure II.

Chapter 8

Evaluation of Parallel Coordinates

Visual perception of graphical representations has been investigated extensively [203], although mostly with 1- or 2-dimensional data. In this chapter[1], eye-tracking characteristics and cognitive performance were investigated in the visual analysis of multi-dimensional data, a common task across many scientific disciplines. The motivation was to find techniques and principles that can be used to design more effective visualizations of multi-dimensional data.

A common strategy for visualizing multi-dimensional data is to use a scatterplot matrix (SPLOM), i.e., a series of 2D Cartesian scatterplots. An alternative strategy is to use parallel coordinates [204], in which data points are represented as polygonal lines intersecting axes in a parallel layout. Both strategies comprise a series of 2D projections that can be used to visually retrieve the value of any data point along any dimension. This retrieval task is a fundamental analytic activity [130] that is a basis for many other compounds or high-level tasks, such as the comparison of single data points or clusters in multi-dimensional space.

It has been shown that scatterplot matrices can be superior to parallel coordinates in judging correlations [205, 206] and in estimating the number of clusters [207], whereas parallel coordinates can be superior in retrieving values

---

[1] **Parts of this chapter have been published in:**
R. Netzel, J. Vuong, U. Engelke, S. O'Donoghue, D. Weiskopf, and J. Heinrich. Comparative eye-tracking evaluation of scatterplots and parallel coordinates. *Visual Informatics*, 1:118–131, 2017 [9], © 2017 Zhejiang University and Zhejiang University Press. DOI: 10.1016/j.visinf.2017.11.001. License: CC BY-NC-ND 4.0.

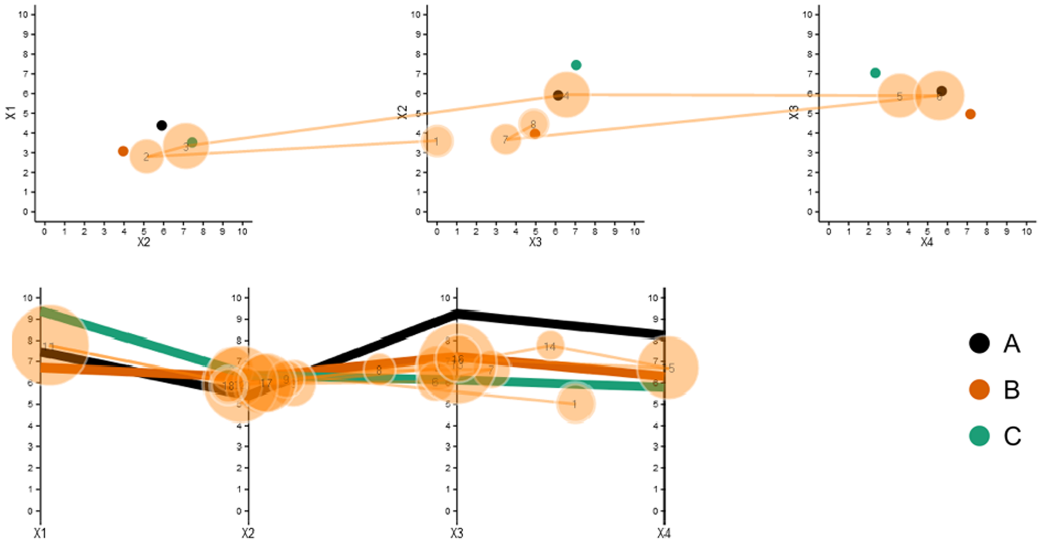


Figure 8.1 — Scanpaths for a study, where a participant was solving a task with 4D data using scatterplots (top) and parallel coordinates (bottom). Given a target point black ('A'), the task was to determine whether the orange ('B') or the green ('C') point was closer to 'A'. With scatterplots, participants focused mostly on the plot interiors and only rarely on the axes. With parallel coordinates, in contrast, participants focused mostly on the axes, especially those at the center of the whole plot.

for coarse data [208]. No study thus has evaluated so far the effectiveness of parallel vs. Cartesian coordinates systems for judging distances between points in multi-dimensional space. Thus, a controlled study was set up in which 24 participants were asked to estimate *relative distances* between points in 2- to 8-dimensional data space (see Figure 8.1). Accuracy, response time, and eye gaze were measured for each diagram type, and for a range of levels of difficulty. Test cases for more dimensions were not included since this would have influenced the size of the plots and thereby the readability of the stimuli. Furthermore, dimension reduction techniques were not considered as an additional factor for visualization and comparison because it is not guaranteed that the projection is stable, possibly resulting in different projections each time the projection is calculated. Therefore, equal distances in the original space could result in different distances in the projected spaces and thereby introduce a confounding factor in this study. By utilizing a SPLOM and parallel coordinates, confounding factors can be avoided in this study which is introduced by the previously mentioned problems. Furthermore, these two techniques can be used without the need to provide additional interaction methods.

In a preliminary study [129] that used similar stimuli and tasks, but non-expert participants, no significant differences in task performance between Cartesian and parallel coordinates were found; however, in the present study participants solved the task faster and more accurately with Cartesian coordinates for data with 2 to 6 dimensions. For 8-dimensional data, parallel coordinates yielded better response times, higher accuracy, and better (subjective) readability regarding intuitiveness, understandability, and clearness of the plot. These results suggest that parallel coordinates may be better suited for comparing data points in higher dimensions.

Besides performance, underlying visual strategies used by participants in estimating the relative distance between points in multi-dimensional data space were investigated. In parallel-coordinates plots, participants focused mostly on the central area of the whole plot and the axes, while in Cartesian scatterplots, participants focused more on the left side of the whole plot and the interior plot areas. These findings provide insights into how participants perceive multi-dimensional data and may be helpful in designing more effective visualizations for such data.

8.1 Related Work

Parallel coordinates are being used more often recently. Therefore, it is of interest how well they perform compared to other well-established visualization methods.

A set of N-dimensional data points can be visualized using either $N-1$ scatterplots [209, 210, 211, 212], or a single parallel-coordinates plot [204]. While in theory, these visualization strategies can convey equivalent information [204], in practice, they transform the data in quite different ways, and it is not always clear which method should be chosen.

In parallel coordinates, axes are drawn in parallel to one another, and each multi-dimensional data point is typically rendered as series of joined, polygonal lines between adjacent axes. The layout order of the axes and/or various parameters of the polygonal lines (e.g., density, color, etc.) can be modified to highlight various aspects of the data as, e.g., reported by Heinrich and Weiskopf [213].

Recently, Johansson and Forsell [214] published a thorough survey of literature on user-centered evaluations of parallel-coordinates techniques, concluding that there is a clear lack of evidence on how to use these techniques most effectively.

Cartesian coordinates (scatterplots) have been found to have advantages over parallel coordinates for some tasks, such as conveying linear correlation within 2D data sets [205, 206], and identifying data clusters [207]. However, parallel coordinates have been shown to outperform Cartesian coordinates for some specific value-retrieval tasks [208].

Other evaluations of parallel coordinates include the one by Kanjanabose et al. [215], who compared user performance across four different tasks using either data tables, parallel, or Cartesian coordinates with a set of eight 4D data points; they showed that the performance depends on the task, with parallel and Cartesian coordinates both outperforming data tables for all tasks except for data retrieval. Additionally, in a more application-driven study, Henley et al. [216] performed a heuristic evaluation of advantages and drawbacks of parallel and Cartesian coordinates for the similarity analysis of genome data.

Eye-tracking measurements can provide insights into the underlying strategies that participants use in solving visualization tasks [149, 148]. Eye tracking has previously been used to study the perception and use of parallel coordinates by Siirtola et al. [217]. Their study compared the recorded eye-tracking data with optimal visual scanpaths and mainly focused on data retrieval tasks that forced participants to look at the coordinate axes. By contrast, the task in this work was quite different, as it focused on estimating relative distances in high-dimensional space; also, the reading behavior of participants was investigated without forcing them to use a specific solution strategy. Furthermore, Siirtola et al. did not perform a statistical evaluation of their results, neither for task performance nor for eye-tracking metrics, both of which are provided in this study. Finally, the design space of this study and that of Siirtola et al. differ considerably, as effects of task difficulty and data dimension are explicitly modeled and evaluated in this study, whereas these factors are hard-coded in a proof-of-concept prototype application that Siirtola et al. used for their study.

8.2 Visual Scanning Model

It was assumed that participants follow some mental model in solving the given task, reflecting possible choices that they can make as well as sequences of actions that are necessary during the execution. In this work, a simplified visual scanning model (see Figure 8.2) of key steps was devised for estimating relative distances in multi-dimensional space with Cartesian or parallel coordinates. This model was used to steer the study design and formulate the hypotheses (see Section 8.3).

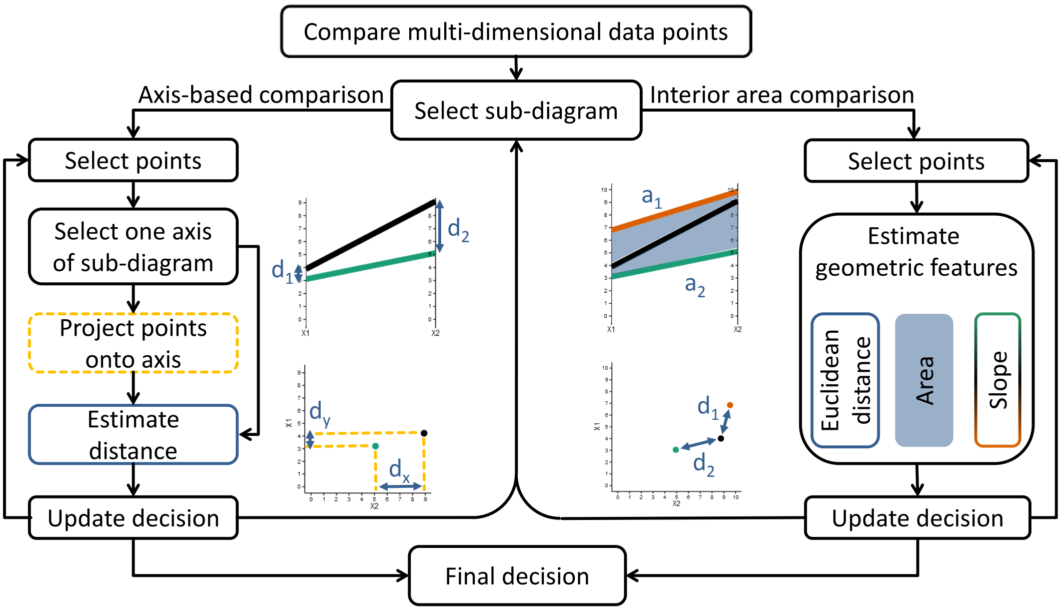


Figure 8.2 — Visual scanning model used in this work: an idealized and iterative process for solving the distance estimation task in multi-dimensional space. Two potentially alternative reading strategies are shown: axes-based comparison (left side) and comparison of interior areas (right side). Distances are denoted by d_x, d_y, d_1, and d_2, as retrieved in either of the sub-diagrams. The areas a_1 and a_2 are compared while inspecting the interior area of the sub-diagram.

Furthermore, it incorporates *elementary cognitive processes* discussed by Simkin and Hastie [218]. They used five basic processes to describe how graphs are compared, which is also reflected in the model:

Anchoring: This corresponds to the *select points* step in the visual scanning model.

Scanning: Switching to another object for comparison. In the model, this can be accomplished by either the inner loop that leads back to the point selection or the outer loop to switch to another sub-diagram.

Projection: Used to connect two points in an image mentally. For *axis-based comparison*, this is part of the step where points are projected onto axes. For an *interior area comparison*, this is used for the estimation of distances.

Superimposing: Moving elements mentally on top of each other. Participants might utilize this during an *estimation* step in the visual scanning model.

Detection: Detecting differences of objects. This is an essential part of the *estimation* steps in the model using either numerical values or geometric features.

More recently, Raschke et al. [88] established an ontology of elements that are relevant in graphs, e.g., they identified axes that participants use (axis-based comparison) and bars (interior area comparison). Based on this ontology, they derived an optimal solution for a given task that they use as a basis to identify and visualize perceptual and cognitive processes.

The model assumes that N-dimensional points are to be visualized in either a single parallel-coordinates plot with axes ordered from 1 to N and left to right, or a series of $N-1$ scatterplots ordered from left to right (see Section 8.3.1 for a detailed description of the stimuli used in this work). The estimation of multi-dimensional distance is modeled as an iterative decision-making process requiring multiple updates until a *final decision* is made by a participant. In each iteration, a participant selects a *sub-diagram*. For Cartesian coordinates, the sub-diagram is an individual scatterplot. For parallel coordinates, the sub-diagram may be a single axis or a plot defined by a pair of neighboring axes. Based on this selection of a sub-diagram, a participant estimates a relative distance either by comparing distances on axes or the interior of a plot (i.e., the total plot area excluding the axes). In axis-based comparisons, a participant has to consider a single axis (for a single-axis sub-diagram) or has to select one of the two axes (for a 2D sub-diagram).

In any case, a participant needs to be able to estimate the distance between two data points on a single axis. For Cartesian coordinates, this requires a participant to visually project the points onto the axis. Then, either an accurate distance could be calculated by reading off values from the axis, or the distance is geometrically estimated along the axis. The same process would then be used to estimate the distance on the second axis (in the case that a 2D sub-diagram was selected). Regardless of whether one or two axes have been considered, a participant now updates the decision (i.e., the current estimate of multi-dimensional distance) based on information extracted from a sub-diagram and proceeds either with the selection of a new sub-diagram or with converting the updated decision to the final decision.

A second way to extract information for the decision-making process is to perform a comparison of geometric features in the interior plot area of a 2D sub-diagram (this second way does not work for single-axis sub-diagrams). For Cartesian coordinates, a participant has to visually estimate the (Euclidean) distance between multiple pairs of points. The visual impressions are then used to update the decision. Parallel coordinates require an estimation of shapes (areas or slopes) formed by pairs of data points. These shapes have to be compared mentally to update the decision of a participant. Again, the process can be continued once a new sub-diagram is selected.

The model allows switching between these two approaches, even while the participant is looping through the overall process of multi-dimensional data comparison, i.e., the participant may change between axis-based and interior area comparisons during the task. However, participants will typically stick to one of the approaches, dependent on the type of diagram: it is assumed that axis-based comparisons for single-axis sub-diagrams are preferred for parallel coordinates, whereas interior area comparisons for 2D sub-diagrams are preferred for Cartesian coordinates. The following eye-tracking experiment is designed to test these assumptions and other aspects of the scanning model.

8.3 Eye-Tracking Experiment

This section describes the details of the study, including design, hypotheses, task, stimuli, and participants.

8.3.1 Design

The design choices presented in this section were inferred from a recently performed preliminary study using Amazon's Mechanical Turk [129] as well as further pilot studies for the eye-tracking part. In the Mechanical Turk study, a distance deviation (see definition below) of less than 5% was not found to be suitable for the task, as the accuracy across all participants was found to be very low. Furthermore, only up to 4D coordinate systems were considered, for which no significant effects on the performance of the participants could be found. Therefore, larger distance deviations and coordinate system dimensions were chosen.

This within-subject user study was designed with the aim to investigate performance in judging relative distances between data points in two different coordinate systems $T \in \{\text{parallel coordinates (PC), Cartesian coordinates (CC)}\}$. For a given number of dimensions $N \in \{2,4,6,8\}$, three data points A, B, and C were presented in each plot and participants were asked to judge whether B or C was closer to A in N-dimensional space (see Figure 8.1). To cover different relative distances from points B and C to the reference point, a *distance deviation* $\delta \in \{15\%, 20\%, 25\%\}$ were introduced for the distance $\overline{AB}$ relative to $\overline{AC}$. The *absolute distance* d of point B to the reference point varied between $d \in \{3,5,7\}$. The range of possible values for every dimension was set to $[0,10]$. The absolute distances were used to determine the location of a point B relative to A. Point C was then derived concerning B by adjusting the distance $\overline{AB}$ according to δ. A full factorial experiment with 24-factor combinations and three repetitions was performed, totaling 72 stimuli. All independent variables and associated values are summarized in Table 8.1.

8.3.2 Hypotheses

The hypotheses are derived from the visual scanning model presented in Section 8.2, the results of a preliminary quantitative study conducted using Mechanical Turk [129], and observations from pilot studies for the eye-tracking experiment.

Hypotheses **H1** and **H2** target the general task performance:

H1: The number of dimensions and the distance deviation has significant effects on response time. With increasing dimensionality, more diagrams need to be consulted, and with increasing difficulty (i.e., decreasing distance deviation) participants need to inspect sub-diagrams more thoroughly. The visual scanning model (Figure 8.2) reflects this by multiple cycles of selecting sub-diagrams and extracting information from there.

H2: Accuracy will decrease with increasing dimensionality and is higher overall for Cartesian coordinates than for parallel coordinates. The visual scanning model incorporates this within the *update decision* step. Here, possibly uncertain extracted information from multiple dimensions have to be merged mentally, which introduces an error that increases with increasing dimensionality.

For Hypothesis **H3**, basic eye-tracking metrics are used. The average saccade length and average fixation duration could be used for inferences of cognitive processing depth and gaze behavior (see Section 2.2.2), which are represented in the *select sub-diagram* step of the visual scanning model (Figure 8.2).

H3: Increasing dimensionality affects the eye-gaze characteristics. Saccade length will increase since it is possible to perform longer jumps between selected sub-diagrams. The fixation duration will decrease since participants will switch more frequently between sub-diagrams to gather information for a fast comparison with a lower cognitive processing depth, which is necessary to update or correct the decision for solving the task.

Table 8.1 — Independent variables used in the eye-tracking experiment.

Independent variable	Values
Coordinate system type T	Parallel, Cartesian
Coordinate system dimension N	2, 4, 6, 8
Distance deviation δ	15%, 20%, 25%
Absolute distance d	3, 5, 7

Hypotheses **H4** to **H6** are based on areas of interest (AOIs) appended to the eye-tracking data. They reflect observations from pilot studies as well as steps of the visual scanning model. In particular, AOIs enable the establishment of correspondence to the visual scanning model and help to investigate the *select sub-diagram* step of the model in more detail. Hypothesis **H4** considers the outer loop of the visual scanning model: checking which sub-diagrams are investigated, thus exploring the distribution of attention on a coarse level. Hypothesis **H5** investigates the distribution of attention on a more fine-grained level by distinguishing between axes and interior parts of the sub-diagrams. Hypothesis **H6** is related to the transition behavior between sub-diagrams.

H4: The focus of the participants will be near the center of the full plot with a tendency toward the left side. This assumption is based on a general bias toward the center of a stimulus (see Holmqvist et al. [36], Chapter 11.6.5; page 397) and also on cultural aspects such as the reading direction of text (see Holmqvist et al. [36], Chapter 11.6.2; page 395).

H5: With Cartesian coordinates, most participants will use the interior plot area to solve the task, whereas most participants will conduct axis-based comparisons for parallel coordinates.

H6: The number of dimensions influences the reading behavior. Transitions between adjacent sub-diagrams will occur more frequently than transitions between non-adjacent (far) sub-diagrams since information from adjacent sub-diagrams will be used mainly for comparison of different dimensions during the decision-making process. The visual scanning model reflects this through multiple possible iterations of the outer loop and the *update decision* step.

8.3.3 Task

Participants were asked to solve a forced-choice task to retrieve the data point closest to the reference point in multi-dimensional space. Participants were instructed to answer as correctly and as fast as possible; no time limit was imposed on them. Note that this low-level distance-estimation task was chosen because it is involved in multiple high-level tasks such as outlier detection or distance estimation between clusters of data items.

8.3.4 Stimuli

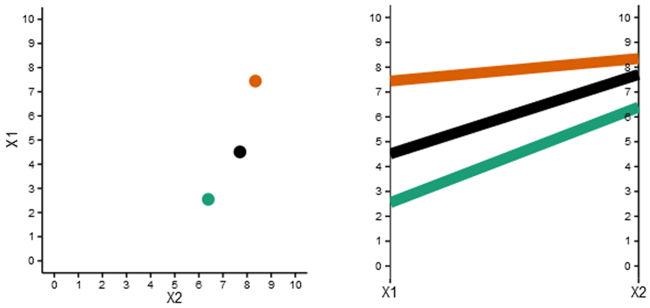
All stimuli were generated by randomly sampling points from the range $[0,10]$, independently for each dimension, using the following steps:

1. The reference point A was placed within the range $[2,10]$ for each axis, ensuring that all points did not intersect with AOIs that mark axes (see Section 8.5.2 for a description of AOIs).
2. Point B was selected via random sampling with an absolute distance of $d_B \in \{3,5,7\}$ to point A in N-dimensional space.
3. Point C was selected by random sampling so that the distance to A was equal to $d_C = (1 + dd) * d_B$; $dd \in \{\pm 0.15, \pm 0.2, \pm 0.25\}$.
4. It was ensured that, in 50% of all cases, B was the correct solution.

Distance deviations were scaled concerning the number of dimensions, using a factor of $\sqrt{N}$. This countered the effect that the projected distances on all axes would become smaller and therefore the task would become more difficult with an increasing number of dimensions and a constant distance deviation.

A single parallel-coordinates plot shows N dimensions using N axes, as well as $(N-1)$ two-dimensional correlations between pairs of adjacent axes (based on the point–line duality [204]). To obtain a fair comparison between both coordinate system types, a linear layout of $N-1$ scatterplots was used in this study, despite the scatterplot matrix (SPLOM) being the most commonly used layout for visualizing N-dimensional data in Cartesian coordinates. Such a linear layout guarantees that the same 1D and 2D features are visible in both plots in the same order and using the same screen real estate. In contrast, the SPLOM layout visualizes N^2 scatterplots (of which $\frac{N(N+1)}{2}$ are redundant), which would either result in less screen space per scatterplot, or a much larger overall stimulus for Cartesian coordinates than for parallel coordinates.

The order of axes was chosen to match exactly between scatterplots and parallel coordinates. While Kuang et al. [208] found that an optimized layout with a single common vertical axis on all scatterplots (which they called *SCP-common*) performed better for tracing tuples, it does not reflect the same information as is



(a) (b)

Figure 8.3 — Example of a single diagram used in this study for Cartesian coordinates (a) and parallel coordinates (b).

shown in the corresponding parallel-coordinates plot due to a different order of axes. Also, the optimal order of axes for parallel coordinates is unknown for the same task, for which it is believed that the chosen layout for scatterplots in this study provides a good balance between common practice and a fair comparison between Cartesian and parallel coordinates.

All stimuli had a resolution of 1920 × 1200 pixels, with diagrams laid out horizontally. Each plot (i.e., 2D correlation) within a stimulus had a size of approximately 270 × 270 pixels. Examples of a stimulus for $N = 2$ including both parallel coordinates and Cartesian coordinates are illustrated in Figure 8.3. Points A, B, and C were colored black, orange, and green, respectively. These colors were chosen from a color-blind safe color table obtained from ColorBrewer.org [219]. Colors were used to enable participants to visually distinguish reference and target points without having to lookup and remember labels. As a side-effect, colors further supported participants in establishing a correspondence of points between dimensions.

8.3.5 Study Procedure

In the beginning, each participant was given information about the study, covering details of the collaboration partners, risks, benefits, how the recorded data would be used, and details about the task they had to solve. Next, they signed the participant consent form, followed by a questionnaire about general demographic information. Vision acuity and color vision deficiencies were tested using Snellen and Ishihara charts. Then, both diagram types were explained in detail, followed by a training phase in front of the eye tracker, which was not recorded. Participants were instructed to press one of two buttons to solve the task: *Space* if point B (orange) was closer to point A (black), and *Enter* if point C (green) was closer to point A. Markers with the respective colors were placed near the keys, as well as at the side of the eye tracker within the periphery of the field of vision. This was to remind the participants which color was associated with which button.

After the training phase was successfully finished, the main study began. First, the eye tracker was calibrated. Next, the participant had to solve the task for 36 stimuli of one diagram type. Finally, the participants were asked to fill out a questionnaire about the visualizations used in the study. The same process was repeated for the other diagram type. The order of these two blocks was counterbalanced: 50% of participants started with parallel coordinates, the other 50% with Cartesian coordinates. The order of the dimensions within each block was also counterbalanced by building four sub-blocks, leading to 24 possible permutations. Each permutation was assigned a unique ID and was used in both blocks. Each sub-block contained nine stimuli. A change of dimension was indicated by a black slide that exhibited the dimensionality of the next shown stimuli. Participants drew an ID at the beginning of the study. Participants were allowed to have a break after they performed the task for half of the stimuli. On average, a whole session lasted about 45 minutes.

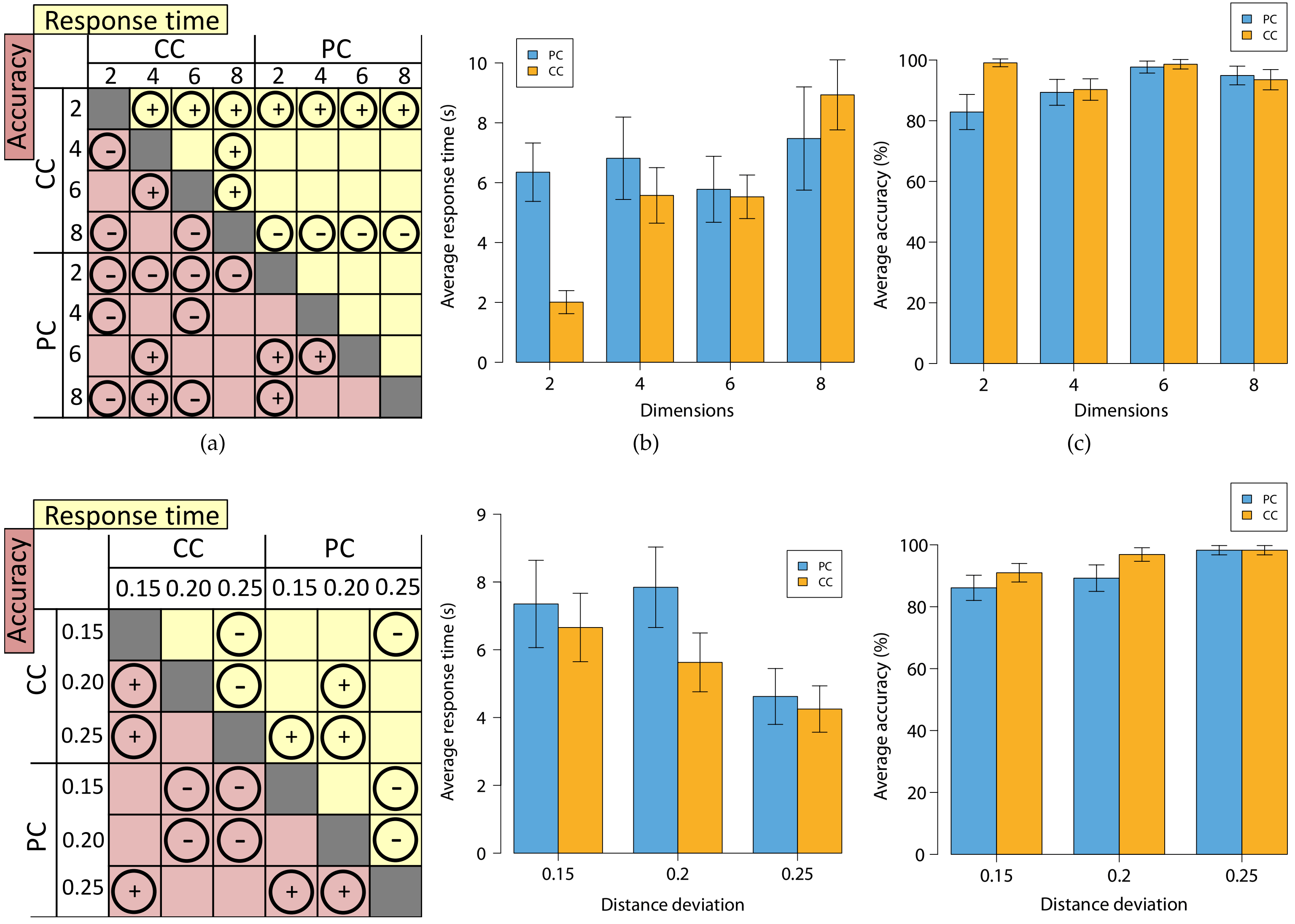
8.4 Task Performance Evaluation

In this section, the results of the analysis of accuracy and completion time are presented.

8.4.1 Response Time

According to the test results, there were significant effects between the number of dimensions ($\chi^2(3) = 14.35$; $p = 0.002$) and the distance deviations δ ($\chi^2(2) = 11.26$; $p = 0.003$) on response time. There was no significant effect for diagram type ($\chi^2(1) = 0.188$; $p = 0.665$). A post-hoc analysis showed significant differences between pairs of dimensions. The results are reported in Figure 8.5a. Additionally, Figure 8.4b shows that response times of Cartesian coordinates (CC) were overall shorter than those for parallel coordinates (PC), except for $N = 8$ dimensions. While response times increased with the number of dimensions, they varied less with parallel coordinates across all dimensions.

Furthermore, significant interaction effects between diagram types and number of dimensions ($\chi^2(3) = 178.89$; $p < 0.001$) were found. Significant results of a pairwise comparison of dimensionality and diagram type are highlighted in Figure 8.4a. Note that response times for 2D and 8D in CC were significantly



		CC 2	CC 4	CC 6	CC 8	PC 2	PC 4	PC 6	PC 8
CC	2		+	+	+	+	+	+	+
CC	4	−			+				
CC	6		+		+				
CC	8	−		−		−	−	−	−
PC	2	−	−	−	−				
PC	4	−		−					
PC	6		+			+	+		
PC	8	−	+	−		+			

(a)

(b)

(c)

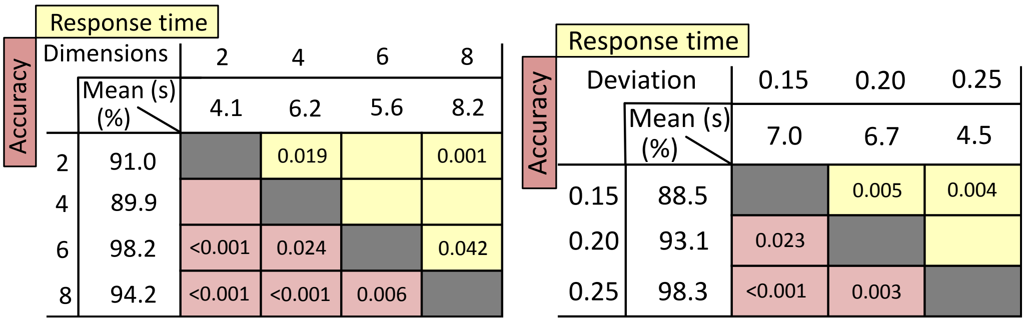
		CC 0.15	CC 0.20	CC 0.25	PC 0.15	PC 0.20	PC 0.25
CC	0.15			−			−
CC	0.20	+		−		+	
CC	0.25	+			+	+	
PC	0.15		−	−			−
PC	0.20		−	−			−
PC	0.25	+			+	+	

(d)

(e)

(f)

Figure 8.4 — Results for the evaluation of task performance. Images (a), (b), and (c) report results with respect to dimensionality, whereas images (d), (e), and (f) show results with respect to the distance deviation. Figures (a) and (d) summarize the results of the statistical tests. A yellow background indicates a comparison with respect to response time and a red background stands for accuracy. Significant differences are highlighted with a ⊕ or ⊝. A ⊕ indicates that the horizontal combination of factors achieved a better result (i.e., shorter response time or higher accuracy, respectively) than the vertical one. A ⊝ stands for an opposite result. The barcharts show mean values. Error bars indicate the confidence intervals calculated based on standard error of the means (SEM).



Response time

Accuracy

Dimensions		2	4	6	8
	Mean (s) / (%)	4.1	6.2	5.6	8.2
2	91.0		0.019		0.001
4	89.9				
6	98.2	<0.001	0.024		0.042
8	94.2	<0.001	<0.001	0.006	

(a)

Response time

Accuracy

Deviation		0.15	0.20	0.25
	Mean (s) / (%)	7.0	6.7	4.5
0.15	88.5		0.005	0.004
0.20	93.1	0.023		
0.25	98.3	<0.001	0.003	

(b)

Figure 8.5 — Results of the post-hoc tests for response time and accuracy with respect to the number of dimensions (a) and distance deviation (b). Mean values are reported as well as p-values, if there is a significant effect indicated by the tests.

different from all other conditions with any number of dimensions. The results of the post-hoc evaluation for the distance deviation are stated in Figure 8.5b. Figure 8.4e shows that, as the task became easier (with increasing δ), the response time decreased for both Cartesian and parallel coordinates.

Here, significant interaction effects between distance deviation and diagram type ($\chi^2(2) = 10.21$; $p < 0.001$) were identified. These are highlighted in Figure 8.4d. Note that $\delta = 0.25$ is the only value for distance deviation that was found to be significantly different from other values of δ.

These results support hypothesis H1: both the number of dimensions and the distance deviation had a significant effect on response times.

8.4.2 Task Accuracy

Similar to response times, significant effects for the number of dimensions ($\chi^2(3) = 28.03$; $p < 0.001$) and distance deviation ($\chi^2(2) = 26.80$; $p < 0.001$) were found considering the task accuracy. Additionally, the diagram type had a significant effect on accuracy ($\chi^2(2) = 7.72$; $p = 0.005$). A post-hoc analysis with respect to the dimensions revealed the results shown in Figure 8.5a and Figure 8.4c.

Pairwise significant interaction effects are depicted in Figure 8.4a. From these figures, it is evident that Cartesian coordinates significantly outperformed parallel coordinates for 2 dimensions. For 4 and 6 dimensions, Cartesian coordinates performed slightly better. For 8 dimensions, parallel coordinates performed slightly better than Cartesian coordinates. Note that for parallel coordinates, the average accuracy overall increased with increasing number of dimensions. Significant interaction effects were found between number of dimensions and diagram type ($\chi^2(3) = 38.09$; $p < 0.001$).

A post-hoc pairwise comparison with respect to the distance deviation is reported in Figure 8.5b. The findings of the post-hoc test are shown in Figure 8.4f. They indicate a linear relationship between distance deviation and accuracy. This is indicated by a linear least squares regression, which results in an intercept of 73.84% and a slope of $97.22\frac{\%}{\delta}$ with a residual standard error of 15.02% on 574 degrees of freedom. Therefore, an increase of the distance deviation by 0.05 resulted in an increase of accuracy by approximately 4.86%. Again, significant interaction effects were indicated between distance deviation and diagram type ($\chi^2(2) = 7.15$; $p < 0.001$). These are highlighted in Figure 8.4d.

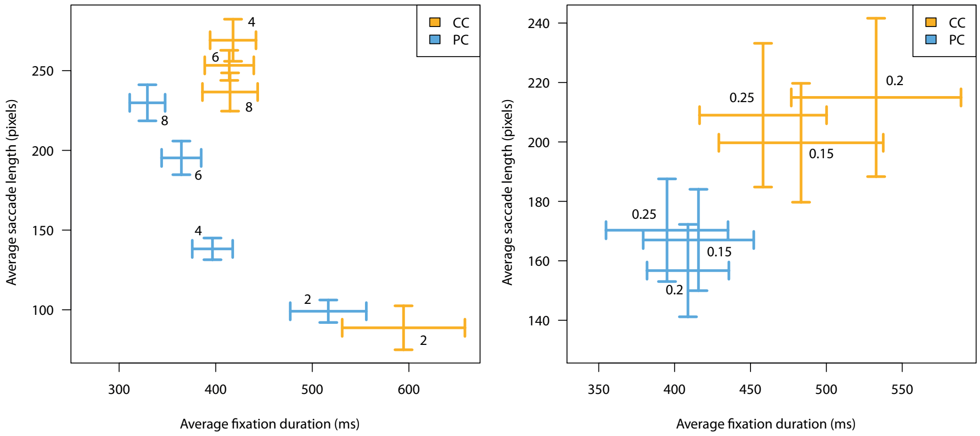
According to these results, H2 can be accepted partially. While accuracy was higher overall for Cartesian coordinates, significant differences were only detected for $N = 2$. Unexpectedly, however, accuracy did not decrease with an increasing number of dimensions. In contrast, accuracy increased from $N = 2$ to $N = 6$ dimensions for parallel coordinates, and was higher for $N = 2$ and $N = 6$ than for $N = 4$ and $N = 8$ for Cartesian coordinates.

8.5 Eye-Tracking Evaluation

In this section, at first, results from analyzing two common eye-tracking metrics (average fixation duration and average saccade length) are presented, followed by an investigation of the AOIs participants were fixating on and the transition behavior between different AOIs. The analysis of these parameters provides insights into possible strategies adopted by participants to solve the given task.

For both eye-tracking metrics, the analyzed distributions did not exhibit a normal distribution but were rather skewed toward one side, which is a typical property of these two metrics (see Holmqvist et al. [36, Chapter 3.3.1; page 88]). Therefore, non-parametric tests were performed. This was also the case for most of the AOI-based analyses.

To improve the quality of the evaluation, the first and last fixation on each stimulus for each participant were removed. These typically represent noise, as most participants exhibit random fixations while processing the start of a new stimulus and, at the end of a trial, eye gaze occasionally left the screen, since participants were required to press a button on the keyboard.



(a) (b)

Figure 8.6 — Average saccade length (y-axis) and average fixation duration (x-axis) for two independent variables: dimension (a) and distance deviation (b). Error bars show the confidence intervals calculated based on standard error of the means (SEM).

8.5.1 Eye-Tracking Metrics

Investigating the average saccade length, significant effects were found for the number of dimensions ($\chi^2(3) = 43.01$; $p < 0.001$) and diagram type ($\chi^2(1) = 16.11$; $p < 0.001$).

A post-hoc analysis revealed significant differences between the following pairs of dimensions: 2-4, 2-6, 2-8, 4-6, 4-8 with $p < 0.05$ in all cases. From Figure 8.6a, it can be observed that the average saccade length increased with an increasing number of dimensions for parallel coordinates, whereas it decreased for scatterplots with $N > 2$. Average saccade length was shortest for 2 dimensions and Cartesian coordinates and longest for 4 dimensions and Cartesian coordinates.

For Cartesian coordinates, the mean average saccade length was 207.89 pixels, and for parallel coordinates, it was 164.67 pixels. The differences concerning distance deviation and diagram type are shown in Figure 8.6b.

Furthermore, there was an interaction effect between dimensions and diagram type ($\chi^2(3) = 260.12$; $p < 0.001$) but not between distance deviation and diagram type ($\chi^2(2) = 4.09$; $p = 0.13$).

Concerning the average fixation duration, significant effects were indicated for the number of dimensions ($\chi^2(3) = 32.59$; $p < 0.001$) and diagram type ($\chi^2(1) = 4.34$; $p < 0.03$).

A post-hoc analysis concerning the number of dimensions revealed significant differences between $N = 2$ and all other dimensions. Figure 8.6a shows that for parallel coordinates, fixation duration decreased with increasing dimensionality, while it was constant for scatterplots with $N > 2$. The average fixation duration was shortest for $N = 8$ in parallel coordinates and longest for $N = 2$ in Cartesian coordinates. The mean average fixation duration for Cartesian coordinates was 491.47 s, and 406.51 s for parallel coordinates.

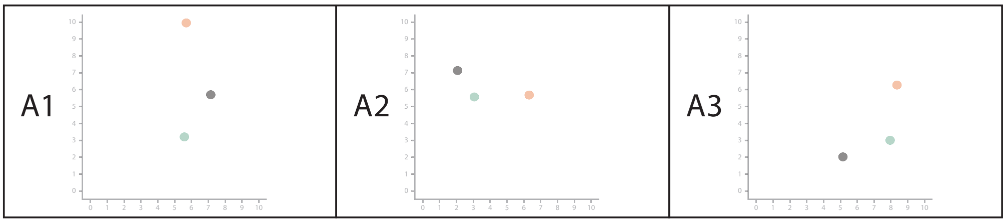
A test for interaction effects shows that there were interactions between dimensions and diagram type ($\chi^2(3) = 19.84$; $p < 0.001$) and between distance deviation and diagram type ($\chi^2(2) = 10.17$; $p = 0.006$).

These results partially support hypothesis H3: with increasing dimensionality, the saccade length increased, and the fixation duration decreased for parallel coordinates, but not for Cartesian coordinates.

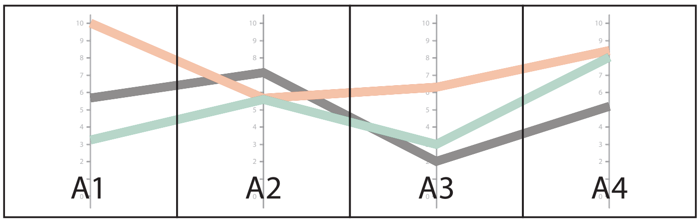
8.5.2 Areas of Interest

To get insight into how participants read parallel and Cartesian coordinates, the distribution of fixations and the transitions between AOIs were analyzed. To this end, a set of *coarse* and *fine* AOIs was defined for both diagram types. For Cartesian coordinates, a coarse AOI contains two axes and the interior plot area of a single scatterplot, whereas for parallel coordinates a coarse AOI contains one axis and half of the interior plot area toward both neighboring axes. This is shown in Figures 8.7a and 8.7b. By defining fine AOIs, the interior plot area was separated from the axes. For Cartesian coordinates, this results in one AOI for both axes and one for the interior plot area of each sub-diagram. The AOIs are defined similarly for parallel coordinates. Each axis is covered by one AOI, whereas the interior plot area between two axes is covered by another AOI. Figures 8.7c and 8.7d illustrate this. With the fine AOIs, it was possible to recognize which path participants followed within the visual scanning model (*axis-based* or *interior area comparison*).

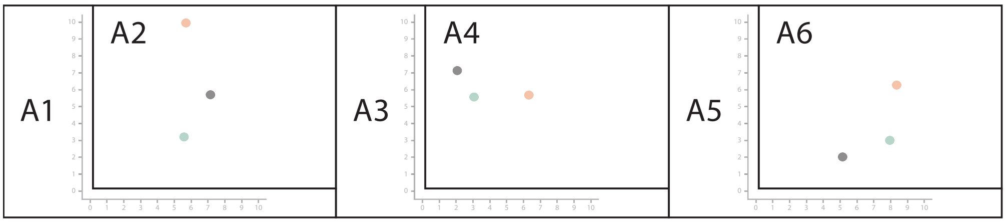
All AOIs are indexed, such that for fine AOIs, odd indexes reference AOIs on axes. To define AOIs without overlap, coarse AOIs in parallel coordinates were centered on axes. As a result, N coarse AOIs were defined for parallel coordinates and $N-1$ coarse AOIs for Cartesian coordinates. Similarly, $2(N-1)$ fine AOIs were obtained for Cartesian coordinates and $N + (N-1)$ fine AOIs for parallel coordinates.



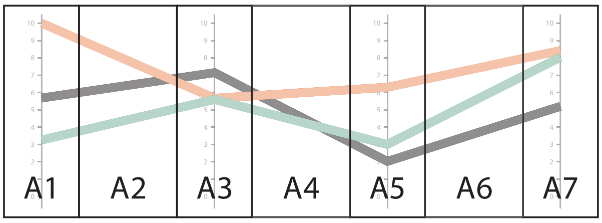
(a) Coarse AOIs for Cartesian coordinates



(b) Coarse AOIs for parallel coordinates



(c) Fine AOIs for Cartesian coordinates

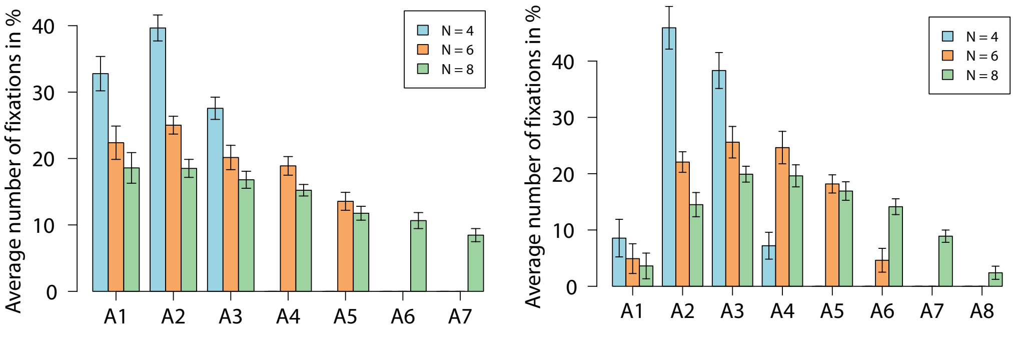


(d) Fine AOIs for parallel coordinates

Figure 8.7 — Coarse ((a), (b)) and fine ((c), (d)) AOIs for 4 dimensions and both diagram types. Coarse AOIs cover a single plot (a) or axis (b), whereas fine AOIs separate axes from interior plot areas.

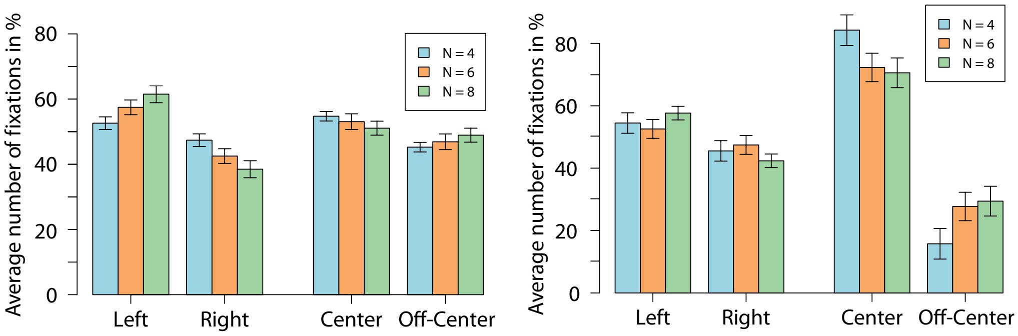
Distribution of Attention

The first part of the analysis concerns the distribution of fixations over the full diagram, using coarse AOIs. The influence of the diagram type and the number of dimensions was of interest. Results for $N = 2$ were discarded since fixation counts are equal to 100% for Cartesian coordinates; for parallel coordinates, it was assumed that attention would split equally between the two AOIs.



(a) Cartesian coordinates (b) Parallel coordinates

Figure 8.8 — Distributions of fixations using the coarse set of AOIs with respect to the dimension and Cartesian coordinates (a) or parallel coordinates (b). Values are given as percentages.



(a) Cartesian coordinates (b) Parallel coordinates

Figure 8.9 — Distributions of fixations separated into left, right, center, and off-center side of the whole diagram with respect to the dimension and Cartesian coordinates (a) or parallel coordinates (b). Values are given as percentages.

The resulting spatial distributions over all coarse AOIs are reported in Figure 8.8. From this figure, the following observations can be inferred: for Cartesian coordinates, the spatial distribution of fixations is skewed, with a preference toward the center/left of the whole plot; for parallel coordinates, the spatial distribution is symmetric with a bias toward the center.

To allow for a statistical test of these observations, coarse AOIs were aggregated and classified as one of two pairs. The first classification pair is left/right. For parallel coordinates, there was always an even number of coarse AOIs, which can be symmetrically assigned to left and right. Cartesian coordinates come with an odd number of coarse AOIs: here, the fixations of the center-most AOI are split to left and right equally. The second classification pair is center/off-

Table 8.2 — Results of the statistical tests for parallel coordinates (PC) and Cartesian coordinates (CC) with a categorization of Left/Right and Center/Off-Center. The degree of freedom is equal to one for all tests and the residuals are equal to 46.

Diagram	Mean (%)		F(1,46)	p-value
	Left	Right		
CC	57.2	42.8	198.5	< 0.001
PC	54.9	45.1	31.1	< 0.001
	Mean (%)			
	Center	Off-Center		
CC	52.9	47.1	44.9	< 0.001
PC	75.1	24.3	258.9	< 0.001

Table 8.3 — Results of the statistical tests for parallel coordinates (PC) and Cartesian coordinates (CC) with a categorization of Center/Off-Center considering the dimensions. The degree of freedom is equal to one for all tests and the residuals are equal to 46.

Dimensions	Diagram	Mean (%)		F(1,46)	p-value
		Center	Off-Center		
4	CC	54.7	45.3	81.3	< 0.001
	PC	84.2	15.8	375.3	< 0.001
6	CC	53.1	46.9	12.8	< 0.001
	PC	72.2	27.7	185.2	< 0.001
8	CC	51.1	48.9	1.9	0.17
	PC	70.6	29.4	144.1	< 0.001

Table 8.4 — Average frequencies of fixation that are located on axes with respect to the dimension and diagram type.

Dimensions	Cartesian	Parallel	$\chi^2(1)$	p-value
2	8.4%	57.8%	34.07	< 0.001
4	10.9%	69.7%	35.27	< 0.001
6	11.8%	67.2%	35.27	< 0.001
8	11.9%	63.8%	35.27	< 0.001

center. The AOIs are classified according to the following constraints: First, both groups should cover equal areas of the whole plot and, second, the center area is spatially centered in the whole plot. Some AOIs have to be split, and their fixations are proportionally assigned to center and off-center.

With these classifications, significant differences between spent attention were found. The results of the statistical test are shown in Table 8.2.

Significant differences are also indicated between center and off-center fixations for both parallel and Cartesian coordinates when taking the number of dimensions into account. In particular, significant differences were found in parallel coordinates for all dimensions. The results of the tests are shown in Table 8.3. The corresponding bar charts are depicted in Figure 8.9. According to these test results, H4 can be accepted.

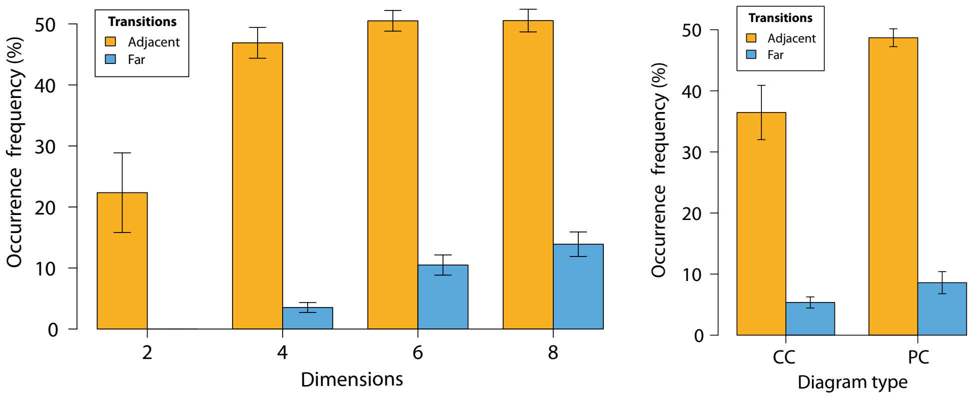
To investigate to which extent participants were using axes to compare values, the frequency of fixations on fine AOIs was counted. The resulting average frequencies of fixation that lie on axes are shown in Table 8.4, ordered by dimensions and diagram types. Overall, this table shows that axes were used significantly more frequently with parallel coordinates than with Cartesian coordinates ($\chi^2(1) = 142.94$; $p < 0.001$). This is also the case for each dimension independently (see p-values reported in Table 8.4). Furthermore, participants tended to focus slightly less on axes with an increasing number of dimensions while using parallel coordinates. For Cartesian coordinates, participants tended to focus slightly more on the axes with an increasing number of dimensions. These results support hypothesis H5.

Transitions

Hypothesis H6 considers the visual scanning strategy regarding transitions between sub-diagrams. Concretely, it is concerned with the spatial span of transitions, i.e., whether transitions are between close-by or distant sub-diagrams.

These kinds of events are well-known and used for analysis in eye-tracking research (see Holmqvist et al. [36], Chapter 6.3.3; page 190).

Here, transition events were classified as *adjacent transitions* if the transitions were between two coarse AOIs that are directly next to each other. In contrast, *far transitions* were between further apart coarse AOIs. Using this classification, a *transition distance* was calculated with two levels: adjacent transitions and far transitions. Based on this classification, counts for the factor transition distance and all other independent variables (diagram type, number of dimensions, and distance deviation) were compared. The counts were given as frequencies relative to the total number of transitions.



(a) (b)

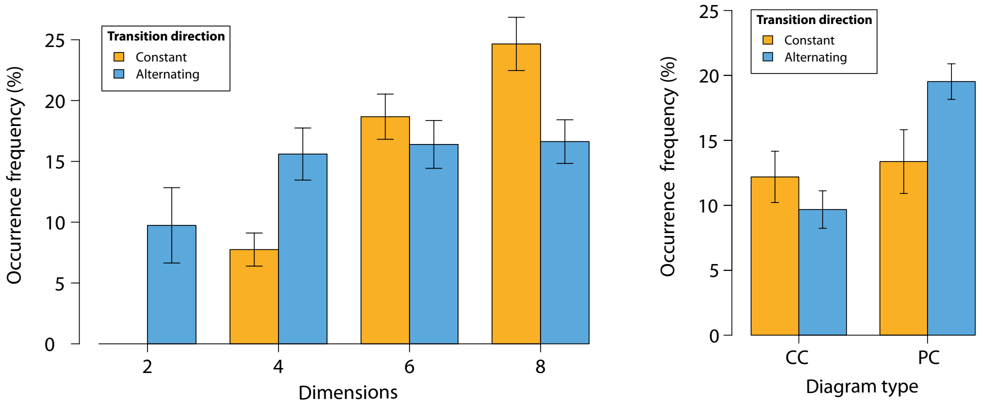
Figure 8.10 — Relative frequencies of far and adjacent transitions between AOIs with respect to the dimensionality (a) and diagram type (b). Error bars show the confidence intervals calculated based on standard error of the means (SEM).

Significant differences were found between adjacent (mean = 42.6 %) and far (mean = 6.9 %) transitions ($\chi^2(1) = 35.26$; $p < 0.001$). Furthermore, there was a significant effect of dimension and transition distance ($\chi^2(7) = 176.93$; $p < 0.001$). A post-hoc evaluation revealed that all pairwise combinations differed significantly except for the combination of 6D/adjacent-8D/adjacent. Mean values are shown in Figure 8.10a.

The combination of diagram type and transition distance showed a significant effect ($\chi^2(3) = 82.24$; $p < 0.001$). Here, a post-hoc analysis showed significant differences for all factor combinations. Mean values are shown in Figure 8.10b. These results support hypothesis H6.

To further investigate the reading behavior of participants, suitable transition events derived from sequences of successive AOIs were analyzed. Using the coarse set of AOIs, possible strategies to solve the task were investigated by extracting occurrence frequencies of associated eye-movement patterns. To this end, two subsequent transitions were considered, i.e., transitions between three consecutive fixations and their associated AOIs.

Sequences of three AOIs were classified into the following two levels of the *transition direction*: For the level *constant transition direction*, the reading direction was constant; subjects performed consecutive eye movements in right or left direction across the three AOIs, thus inspecting consecutive diagrams. An *alternating transition direction* occurred when the reading direction changed from left to right or vice versa, indicating that participants had to compare diagrams frequently.



(a) (b)

Figure 8.11 — Relative frequencies of constant and alternating transition directions with respect to the dimensionality (a) and diagram type (b). Error bars show the confidence intervals calculated based on standard error of the means (SEM).

Next, counts were compared for the factors transition direction, diagram type, and number of dimensions. The counts are given as frequencies relative to the total number of 3-AOI-sequences. Based on this sequence analysis it was tested whether the reading direction was kept or whether it changed. Significant difference were found for counts between alternating and constant transition directions ($\chi^2(1) = 4.34$; $p < 0.037$). The mean for constant transition direction was 12.8%, and for alternating transition direction, it was 14.6%. Both diagram types ($\chi^2(3) = 44.08$; $p < 0.001$) and number of dimensions ($\chi^2(7) = 138.62$; $p < 0.001$) had a significant effect. A post-hoc evaluation for diagram type and transition direction revealed significant differences between all combinations of factors, except between Cartesian coordinates and parallel coordinates concerning constant transition direction. For the number of dimensions, differences were found between all combinations of factors, except for two sets of combinations: First, there was no significant difference between constant and alternating transition direction for $N = 6$. Second, $N = 8$ and alternating transition direction was not significantly different from $N = 4$ and $N = 6$ for alternating transition direction as well as from $N = 6$ for constant transition direction (see Figure 8.11). These results indicate the composition of the *adjacent* and *far transitions* regarding *constant* or *alternating transitions*. Using parallel coordinates, participants changed their reading direction more frequently than while using Cartesian coordinates. Regarding the dimensions, there seems to be a trend toward *constant transitions* with increasing dimensionality, while the frequency of *alternating transitions* seem to stay constant.

8.6 User Feedback

In this section, results from the questionnaire given to participants during the experiment are presented.

Three questions were asked: First, participants should state — according to their opinion — the percentage of stimuli for which they believed they had solved the task with absolute certainty. For parallel coordinates, the average percentage was 68.5%, and for the Cartesian coordinates, it was 80.4%. This was lower, on average, for both diagram types compared with the average task accuracy of over 90%.

In the second question, participants were asked to write down the strategies they used to solve the task. This question should have helped identify which paths the participants have taken within the visual scanning model to solve the task. For Cartesian coordinates, there were three answers: compare the differences visually (chosen by 9 participants), calculate the differences (3 participants), and estimate by majority vote (9 participants). For parallel coordinates, the following strategies were chosen: estimate the distance by visual means (8 participants), calculate the overall distance by summation of single differences (4 participants), compare the areas between the black line and the others (4 participants), and again perform a majority vote (7 participants). Note that not all participants have answered this question; some stated instead that they had no particular strategy. Overall, this supports the finding that participants tended not to use axes to solve the task with Cartesian coordinates. In the last question, participants were asked to rate the readability of the diagram depending on the number of dimensions with a grade between 1 (best) and 6 (worst). For Cartesian coordinates, average grades worsened with increasing dimensionality from 1.0 to 3.0. For parallel coordinates, grades did not vary much, but worsened from 2.15 to 2.65. Average grades are shown in Table 8.5.

Overall, there were no significant differences ($\chi^2(1) = 2.39$; $p < 0.12$) between

Table 8.5 — Average grades (from 1 to 6, where 1 is best) for the readability of diagrams by number of dimensions.

Dimensions	2	4	6	8
Cartesian coordinates	1.00	1.79	2.29	3.00
Parallel coordinates	2.15	2.27	2.29	2.65
p-values	< 0.001	0.07	0.60	0.17
$\chi^2(1)$	22.39	3.39	0.27	1.85

mean ratings for Cartesian coordinates (mean = 2.02) and parallel coordinates (mean = 2.34). Testing each N individually, a significant difference was only found for $N = 2$.

8.7 Discussion and Guidelines

In the evaluation of the data from this study, a similar approach is taken as in Chapter 6. Task performance is evaluated statistically. Additionally, AOIs are used to get insights into behavioral patterns of participants, as depicted in the workflow of Figure II. Here, the difference is how the AOIs are utilized. Instead of considering basic dwell times, AOIs are used to count how often AOI transitions occurred. These frequencies are then evaluated statistically.

The analysis of response time showed that there was no significant difference overall between diagram types in solving the task. As expected (H1), the complexity of the task (concerning the number of dimensions and the distance deviation) has a significant effect on response time. Furthermore, response time increased with the complexity of the task, while the number of dimensions seemed to have a smaller effect on parallel coordinates than on Cartesian coordinates. For $N = 8$, the task was solved faster with parallel coordinates than with Cartesian coordinates (though not significantly), and for $N = 2$ Cartesian coordinates outperformed parallel coordinates significantly.

In contrast to previous work [129], participants were able to solve the task accurately in most cases, while being significantly more accurate with Cartesian coordinates than with parallel coordinates (H2). This could be the case due to the lower complexity of the task in this study. Note that a high task difficulty was not considered in the study design and as a result, the stimuli have been generated based on the parameters described in Sections 8.3.1 and 8.3.4.

Concerning completion time, the results indicate that parallel coordinates were slightly advantageous over Cartesian coordinates for $N = 8$, while accuracy increased with distance deviation as expected. An indication of the same trend was found for the number of dimensions and parallel coordinates: the results suggest that accuracy may *increase* with the number of dimensions (Figure 8.4c). Note that the distance deviation was adjusted concerning the number of dimensions. This could have affected the accuracy. The visual scanning model reflects this outcome, since with increasing dimensions more iterations are necessary for comparison. Hence, the completion time will increase. The accuracy, on the other hand, should decrease, since each comparison and *decision update* introduces a certain accumulating amount of uncertainty. The paths that can be selected within the model also influence the performance. A comparison of geometric features is faster than utilizing the exact values of the axes, but also more error-prone.

> **Guideline 1** – Cartesian coordinates should be used to represent lower-dimensional data, whereas parallel coordinates should be used for higher-dimensional data.

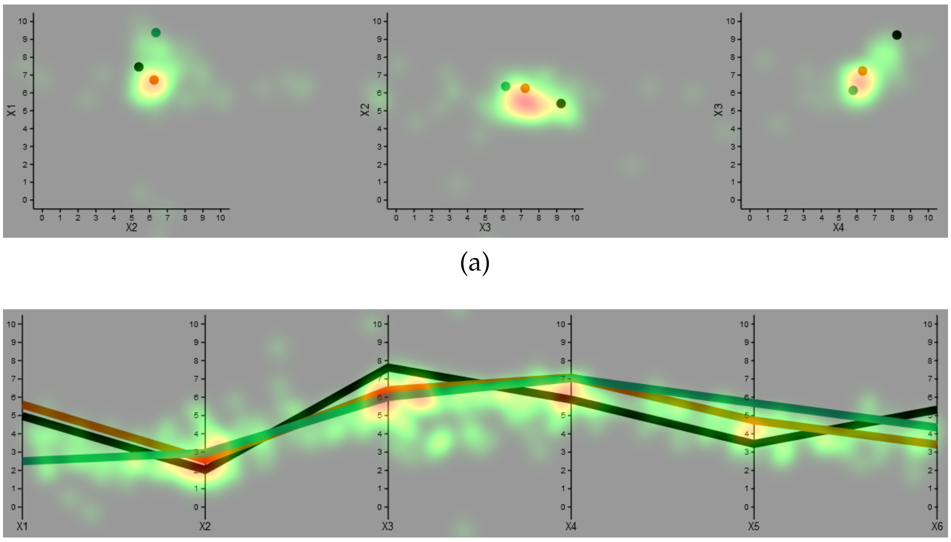
This study motivates a break-even point around $N = 8$. However, the exact dimensionality may depend on several factors and, thus, there is no recommendation for a specific number of dimensions that parallel coordinates should be used for.

The eye-tracking data as presented in Section 8.5 revealed an increasing average saccade length for large N and parallel coordinates (H3), indicating that participants looked at sub-diagrams that are further apart. In combination with a decreasing average fixation duration, this indicates an information retrial from sub-diagrams and a comparison of the retrieved information.

For Cartesian coordinates, two clusters emerge: $N = 2$ vs. $N = 4$, $N = 6$, and $N = 8$ (see Figure 8.6a). This shows that subjects performed jumps of almost equal length between diagrams. A possible explanation for this result is the consistent visual mapping of parallel coordinates: with more dimensions, more axes are added to the *same* plot, whereas new diagrams need to be added for Cartesian coordinates.

> **Guideline 2** – For 2 dimensions, eye-tracking and performance metrics indicate a special case where Cartesian coordinates outperform parallel coordinates. Therefore, Cartesian coordinates should be used for 2 dimensions.

Regarding the spatial attention of participants spent on diagrams, the findings support H4. While participants focused on the center of a parallel coordinates plot, the distribution of fixation counts was skewed to the outer/left for Cartesian coordinates of $N = 4$ or more dimensions. This may be an important result with implications for the design of axis-order algorithms, for both scatterplots and parallel coordinates. As expected, the left side of the stimuli was viewed more often than the right side. The heatmaps in Figure 8.12 illustrate this. It is assumed that this was due to the population sample containing mostly participants from Western countries, where the most common reading direction is left-to-right.



(a)

(b)

Figure 8.12 — Heatmaps aggregated over all participants for Cartesian coordinates and 4 dimensions (a), and for parallel coordinates and 6 dimensions (b). The red spots show that axes were mainly used in parallel coordinates, while participants were judging relative distances in Cartesian coordinates using the interior part of the plot.

These results indicate that the most important axes should be placed around the center of the parallel coordinates contrary to the common principle to place the important axes from left to right, which is based on the assumption that the first dimension is the most significant one [220]. According to a used metric, the significance of the following axes is then decreasing. This concept is, e.g., used to create a *Radviz* [221] visualization to highlight clusters, where dimensions are added greedily. There are also approaches for parallel coordinates that highlight group structures by an automatic selection of the dimension order [222]. The best-suited dimensions are then used as axes placed from left to right. In contrast to automatic methods for the axis-order selection, there are also approaches based on a manual selection of axis-pairs [223]. Here, all dimension pairs are ranked according to different metrics and then displayed from left to right.

> **Guideline 3** – More important axes should be placed near the center of a parallel-coordinates plots. For Cartesian coordinates, these axes should be placed in reading direction.

Furthermore (H5), the data confirms that participants made use of axes more frequently for parallel-coordinates plots than for scatterplots (see hotspots in Figure 8.12). This is consistent with the collected feedback: most participants were solving the task either visually or by a majority vote, where reading off values was not required. Similarly, given that only four participants compared areas in parallel coordinates (which does not require axes) to solve the task, it was expected that the larger fraction of participants would use the axes in parallel coordinates.

> **Guideline 4** – Areas between adjacent axes in parallel coordinates are rarely viewed. Therefore, this space could be used to display additional information or layout line segments in a more elaborate way.

Similarly, it was confirmed (H6) that transitions occurred more frequently between adjacent plots (in Cartesian coordinates) or pairs of axes (in parallel coordinates) than between non-adjacent plots or pairs of axes. This is shown in Figure 8.10. The results indicate that participants followed a strategy to arrive at a conclusion with occasional jumps between non-adjacent plots. These jumps could be explained by features in the stimuli that stand out, such as single AOIs exhibiting very large or very small distance deviations.

Participants changed reading direction frequently, in particular when solving the task with parallel coordinates. This might have been the case since participants had to compare axes frequently during their decision-making process. Regardless of the applied strategy, participants were more likely required to compare axes and update their vote accordingly. The frequency of changes in reading direction correlated with the number of dimensions. This is shown in Figure 8.11a: with increasing dimensionality, the occurrence of consecutive eye movements increased, whereas changing the reading direction in-between from left to right seemed to be constant for $N > 2$. This is shown in Figure 8.11a: with increasing dimensionality, the occurrence of consecutive eye movements increased, whereas changing the reading direction in-between from left to right seemed to be constant for $N > 2$.

As with most user studies, the results and guidelines of this study are only valid within the tested parameter space. Although the number of independent variables was extended compared to previous work, further studies are required to test more realistic scenarios, e.g., by testing even higher dimensions, incorporating the effects of positive and negative correlation and axis order (which is seemingly influencing the ability of subjects to establish a correspondence across multiple sub-diagrams), or by adding noise to the stimuli. Color is also a feature that supports subjects in establishing correspondence between sub-diagrams. In this experiment, the use of colored data points was slightly in favor of Cartesian coordinates, since there was no other visual connection between the sub-diagrams. For parallel coordinates, the line segments already supported establishing a connection between sub-diagrams in addition to the use of color.

Additionally, regarding fairness aspects, participants may not have been used to SPLOM visualizations or parallel coordinates, which was addressed by a sufficient training phase. In the context of fairness, the order of the axes is also relevant, since it could favor either Cartesian coordinates or parallel coordinates in a way that, e.g., distinct visual patterns are always visible in one diagram type and not in the other one. In this experiment, the data dimension did not have a specific meaning or importance through a randomized construction process. Therefore, the order of the axes was not optimized, and only a fixed axis order was used for both Cartesian coordinates and parallel coordinates. Here, an axis order proposed by Kuang et al. [208] was used.

Furthermore, it would be interesting to have a closer look at how expert users work with parallel coordinates in contrast to novice users to extend the validity of the proposed guidelines.

The results also indicate that parallel coordinates might be better suited for sensitivity analysis tasks than Cartesian coordinates since the accuracy for higher dimensions seems to be similar, but the task was solved on average faster. The verification of this assumption needs to be tested in additional experiments.

Another aspect that needs further testing concerns the number of data points. In this study, only three points were used, neglecting possible overdraw of points or intersecting polylines with an increasing number of data points that could lead to visual clutter. However, with an increasing number of data points, individual data points become less relevant, while group structures gain in relevance. In this case, just a few groups may be compared by an analyst, resulting in a scenario similar to the task: How close or similar are the groups in multi-dimensional space?

CHAPTER 9

Conclusion

In this thesis, different approaches for a vision-based analysis of data were presented. Here, an automatic data analysis of image-based representations as well as an analysis based on visualization frameworks is considered. For the data analysis of image-based representations, perceptual models were used as a basis for an adaptation of an image, while the visualization frameworks focused on eye-tracking data to find behavioral patterns. In contrast to these kinds of approaches, this thesis also addressed an empirical hypotheses-driven evaluation of conducted eye-tracking studies to confirm behavioral patterns. A summary of the concepts, an assessment of their utility, and directions for future work building on them is given in the following.

9.1 Summary of Chapters

Methodologies for evaluation are used across all domains, although their expenditure can vary tremendously, ranging from simple workflows utilizing rather basic visualizations to more sophisticated methods using models or more complex systems that follow the concept of *Visual Analytics*. This topic was addressed in the first half of Part I of this thesis.

Models can be used in different ways. As addressed in Section 3.2, models can be utilized for quantification. Here, frequency key values were computed based on a spectral analysis of images generated using different advection methods. With these key values, it is possible to identify the dependency between decreasing radii and blurring of the advected images. Overall, the evaluation with frequency key values showed that the BFECC approach with third-order interpolation generates highly accurate advection textures but suffers from low FPS rates. The backward integration with fifth-order interpolation tends to result in similar texture quality and higher FPS rates. In general, this approach exhibits the best tradeoff between computational speed and texture quality. Another way to utilize models was discussed in Section 3.3. Here, Gabor filters were used as a model for human perception of direction and frequency in addition to models that reflect contrast perception. These models were used in a compensation process aiming to reduce the effect of the simultaneous orientation contrast. Although the proposed process shows promising results, it so far operates only on gray-scale images, neglecting the effects of color perception, which would be a meaningful extension.

For the example of eye tracking, three aspects were identified that need to be addressed, simplified, or considered for evaluation, which is subject to the second half of Part I of this thesis. The first issue was addressed in Section 4.2. Annotated data is the basis for many evaluation frameworks, but the annotation is often a time-consuming and ambiguous process. To overcome these issues, a highly flexible approach was proposed based on manual annotation, where an annotator can utilize visual context for the annotation. The drawback is the time-consuming annotation process itself. Potential improvements could include (semi)-automatic labeling, advanced computer vision techniques for object recognition and tracking, or pattern recognition of the scanpath for automatic fixation estimation. However, any of these extensions would lead to a more specific tool that would lose its generality and flexibility.

Spatiotemporal visualization of eye-tracking data is another topic that needs to be addressed. Currently, a viable popular option to analyze spatiotemporal behavior is based on 3D space-time cubes, which can be used for finding attentional synchrony, but they require interaction methods and are therefore not considered as a static visualization. Additionally, 3D visualizations often suffer from occlusion. Hence, the 2D spatiotemporal visualization in the form of the *Hilbert attention map* was proposed in Section 4.3. The method is based on a projection from 3D space to 2D using a space-filling Hilbert curve supporting brushing-and-linking to incorporate traditional visualization techniques for gaze data.

An important topic for most evaluations is the selection of appropriate metrics. A variety of metrics can be used in most domains, but the question is which ones are relevant in a given scenario. A workflow was proposed to address this topic in Section 4.4 that supports the selection of metrics and also the generation of meta metrics that are in this case an affine combination of a subset of metrics.

Furthermore, a dimensionally stacked similarity matrix (DSSM) was used to analyze group structures visually. Although this method supports an analyst in investigating possible metrics, it is subject to limitations. The display space is one example. More metrics lead to more needed display space. For the DSSM this is a problem due to a quadratic growth of area concerning the number of participants. Nevertheless, there are other ways to visualize metrics like parallel coordinates that only have a linear increase of area per metric. Color encoding of metrics is another issue since it becomes more difficult to select distinguishable colors with increasing number of metrics.

In Part II of this thesis, empirical evaluations were performed in a variety of domains. The focus of the evaluations was on eye tracking and to use recorded eye-gaze data to deepen the understanding of how tested visualizations and methods were utilized and how they affect people concerning performance and behavioral patterns. For this purpose, the evaluations contained basic statistical tests on task performance (accuracy and time) and eye-tracking metrics (fixation duration and saccade length). In further steps, specialized methods were utilized to analyze the data, i.e., deriving descriptive factors from the data, using AOI-based methods, or applying automatic processing in the sense of machine learning.

For the analysis of the encoding of trajectories, only the basic evaluation was performed, but still, clear patterns emerged: According to the results of this study, tapered links performed very well, followed by standard arrowheads for tracing of paths and identifying the longest links. It was also shown that these two link types could lose their ability to encode direction, if interactions, such as zooming, are provided. In this case, equidistant comets and arrowheads benefit from repeatedly placed direction information on a link, which is present at different zoom levels. It was also confirmed that using point splatting increases accuracy.

The results of the basic evaluations of different visual search support concepts showed that each of the three approaches (miniature, directional, and grid reference annotation) leads to significantly reduced completion times. Significant differences between the three annotation methods were found: the lowest and therefore the best completion times were achieved by using miniature annotations, followed by grid-reference annotation, and directional annotation. Correctness was not evaluated since the task could be solved accurately in any case. By evaluating the eye-tracking data and newly derived factors, it could be shown that participants solved the task in two phases: First, they used the provided annotation to get a hint for the position of the label; then, they performed the actual search. Furthermore, the fixation duration changed after the transition into phase two. For phase one, grid-reference annotation had the highest average fixation duration, indicating deep cognitive processing. In phase two, the fixation durations are approximately the same, since similar search is performed for all annotation techniques. There was also a difference in the search behavior: Directional annotation triggers search along straight lines, whereas the other techniques lead to an area-oriented search.

The analysis of metro maps and possible reading behavior was based on task performance and eye-tracking metrics. Furthermore, the analysis showed that color-encoded maps outperform the gray-scale maps. Finally, an analysis of the reading behavior of metro maps was performed based on labeled fixations, machine learning, and the bimodality coefficient. In this way, different strategies could be identified and verified. Furthermore, there are differences concerning the map complexity, task difficulty, and also different clustering groups. The available data indicates that the reading strategies do not depend on the color-encoding—the same kind of strategies are used in both color and gray-scale maps. The results of this study were later qualitatively reproduced [10], utilizing word-sized graphics of common eye-tracking visualizations.

The last chapter in Part II provides an in-depth comparison of parallel coordinates and scatterplots for distance estimation in multi-dimensional space. Concerning task completion time, parallel coordinates performed better for higher dimensions compared to Cartesian coordinates. Cartesian coordinates achieved higher accuracy for lower dimensions, but parallel coordinates were more accurate with the highest data dimension. Different patterns were identified regarding the fixation duration and saccade length: Cartesian coordinates exhibited two clusters of similar behavior. For parallel coordinates, there were no clusters and gaze behavior changed with dimensionality. Another important finding of this study was a skewed distribution of the fixation count to the outer/left for Cartesian coordinates, while for parallel coordinates the center was focused more often, which was derived from an AOI-based evaluation. Furthermore, subjects utilized the axes more frequently for parallel coordinates than for Cartesian coordinates. Additionally, the reading direction changed more frequently when participants were using parallel coordinates. The observations also support the established visual scanning model that describes the use of Cartesian and parallel coordinates on an abstract level. Based on these results, design guidelines were suggested for choosing parallel or Cartesian coordinates as representations, the placement of important axes, and utilizing the space between parallel coordinate axes.

9.2 Future Work

The proposed concepts and methods in this thesis cover only a few approaches for improving visualizations based on perceptual models. Therefore, it would be of interest to further work on adaptive methods to reduce illusionary effects that often appear in image-based representations and might change the meaning of the represented data. For such approaches, eye-gaze data could be utilized during an adaptation or compensation. In the sense of *foveated rendering*, it would be of interest to perform the compensation of such effects locally, at the position of the current eye gaze, and to test its effectiveness.

Another field that is addressed in this thesis is the use of eye-tracking data to extracting behavioral patterns. To this point, alternative ways are presented to analyze user behavior by introducing evaluation schemes that are related to the used stimuli and tasks. Nevertheless, there are still topics that are of interest for further research. It would be interesting to improve further the method of exploring the metric space regarding an interactive analysis of eye-tracking data with the focus on extracting behavioral patterns by, e.g., an automatic analysis of the eye-tracking data resulting in a suggestion for a set of suitable metrics.

Empirical evaluation was another part of this thesis since there seems to be a lack of properly conducted evaluations of eye-tracking studies in a variety of domains. Statistical testing is commonly used to evaluate measurements obtained from user studies in contrast to interactive analysis methods to confirm behavioral patterns. To further address this topic, additional studies could be conducted to investigate or extend the parameter space for the conducted studies. Related to the study of trajectory encoding methods in Chapter 5 that are essentially node-link diagrams, it would be of interest to test the usefulness of partial links since by using them, visual clutter can be reduced, which is often a problem in visualizations of trajectories. Furthermore, a meta-study could be conducted that evaluates analysis or evaluations methods concerning human vision.

In addition to such kind of studies, evaluations schemes could be extended and improved. Instead of using, e.g., common AOI sequence plots to find behavioral patterns, distinct AOI transitions could be defined, and their frequencies could be statistically evaluated.

9.3 Overarching Discussion

We are living a time where much data is being produced increasingly fast. Therefore, it is no longer possible to handle the data manually in many areas. Especially, when it comes to the evaluation of quality, which is not only of relevance in industrial areas but also in academic research. Here, an unbiased assessment is of utmost importance to ensure comparability and reproducibility of results, which can be achieved by quantifying relevant properties. Quantification can be achieved by different means using, e.g., models or metrics. It can be applied to, or incorporated in, a variety of adaptive algorithms, interaction methods, or applications.

In this thesis, methods for two research fields related to human vision were considered for quantitative evaluation and adaptation of visualizations. The first field addresses the adaptation. In visualization, artifacts are often visible that are not present in the data itself but rather introduced by the human visual system, resulting in a possible misinterpretation of the data. Therefore, one aspect of Part I of this thesis focused on an automatic data analysis of image-based representations. Here, perceptual models were used to quantify visual effects and compensate for them to steer the generation of an image-based representation of data to achieve, e.g., a higher visual quality or to represent data more truthfully.

The second field addresses the evaluation, which was here based on recorded eye-tracking data. Such data is becoming more popular in many communities, e.g., in human-computer interaction where methods are developed that utilize recorded eye-tracking data for interaction purposes. The focus of this thesis is on visualization and the use of eye tracking in this field, which is also a growing research area. The main topic here is to provide an effective communication of recorded eye movements and data derived from them to reveal how stimuli are visually accessed and which information is processed by observers. The analysis or evaluation of such data is not an easy task, and standardized practices for technical implementations and data interpretation are still missing.

To this point, another aspect addressed in Part I of this thesis is related to a user-driven examination of eye-tracking data. Here, analysts are confronted with a large amount of data that needs to be surveyed. Unfortunately, there is no general automatic examination method that can be utilized, since an analysis highly depends on the used stimuli and tasks, especially, if it is based on metrics. Therefore, user-driven data examination methods were presented using interactive visualizations to support analysts.

Three major areas were addressed that are of importance in any research of eye-tracking data: annotation of eye-gaze data, spatiotemporal data analysis, and metrics-based analysis. The latter two fields are used to derive behavioral patterns or to identify groups of participants exhibiting similar behavior concerning their eye-movements.

One problem of many analysis methods is their qualitative nature that makes it hard to compare research results. Therefore, it is important to find ways for communicating results in a meaningful way to establish the validity and scope of acquired knowledge or developed methods. To address this issue, quantitative approaches can be used.

Part II of this thesis focused on the empirical evaluation of recorded eye-tracking data to quantify task performance and, more importantly, to confirm behavioral patterns. Here, traditional null hypothesis testing is used in addition to concepts of *new statistics*. Additional information about the data is provided, such as confidence interval or kinds of data distributions, to let a reader interpret the significance of results.

Studies were conducted or evaluated concerning four domains with a variety of different tasks: trajectory visualization, correspondence visualization, automatic extraction of groups of participants with similar eye-movement patterns using maps, and, finally, the comparison of diagram types.

Some general conclusion can be derived from the data analysis and the empirical evaluation of eye-tracking data from this thesis. First, visualizations often do not incorporate human perception, resulting in representations that do not represent the used data truthfully. To this end, this thesis showed that by even using simple methods that build on results from other fields of research, e.g., physiological psychology, representations of data can be improved through adaptation. Similarly, analysis approaches that are easy to use are also quite effective. Although a manual annotation is a time-consuming process, a framework for the manual annotation of eye-gaze data with few basic interaction methods is already sufficient to overcome ambiguity problems like gaze data located in multiple overlapping AOIs or the insufficient accuracy of an eye tracker in combination with small AOIs. For the spatiotemporal analysis of eye-tracking data, instead of utilizing a space-time cube and necessary interactions, a basic 2D scatterplot using space and time as axes in combination with a kernel density estimation can be used to visualize patterns, assuming that space can be projected onto a continuous axis.

Author's Work

[1] R. Netzel, M. Ament, M. Burch, and D. Weiskopf. Spectral analysis of higher-order and BFECC texture advection. In *Workshop on Vision, Modeling and Visualization (VMV)*, pages 87–94, 2012.

[2] R. Netzel and D. Weiskopf. Hilbert attention maps for visualizing spatiotemporal gaze data. In *IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)*, pages 21–25, 2016.

[3] R. Netzel, M. Burch, and D. Weiskopf. Interactive scanpath-oriented annotation of fixations. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ETRA '16, pages 183–187. ACM, 2016.

[4] A. Kumar, R. Netzel, M. Burch, D. Weiskopf, and K. Mueller. Multi-similarity matrices of eye movement data. In *2016 IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)*, pages 26–30, 2016.

[5] A. Kumar, R. Netzel, M. Burch, D. Weiskopf, and K. Mueller. Visual multi-metric grouping of eye-tracking data. *Journal of Eye Movement Research*, 10 (5), 2018.

[6] R. Netzel, M. Burch, and D. Weiskopf. Comparative eye tracking study on node-link visualizations of trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2221–2230, 2014.

[7] R. Netzel, M. Hlawatsch, M. Burch, S. Balakrishnan, H. Schmauder, and D. Weiskopf. An evaluation of visual search support in maps. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):421–430, 2017.

[8] R. Netzel, B. Ohlhausen, K. Kurzhals, R. Woods, M. Burch, and D. Weiskopf. User performance and reading strategies for metro maps: an eye tracking study. *Spatial Cognition & Computation*, 17(1-2):39–64, 2017.

[9] R. Netzel, J. Vuong, U. Engelke, S. O'Donoghue, D. Weiskopf, and J. Heinrich. Comparative eye-tracking evaluation of scatterplots and parallel coordinates. *Visual Informatics*, 1:118–131, 2017.

[10] F. Beck, Y. Acurana, T. Blascheck, R. Netzel, and D. Weiskopf. An expert evaluation of word-sized visualizations for analyzing eye movement data. *IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)*, pages 50–54, 2016.

[11] M. Burch, R. Woods, R. Netzel, and D. Weiskopf. The challenges of designing metro maps. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, volume 2, pages 195–202, 2016.

[12] N. Rodrigues, R. Netzel, K. R. Ullah, M. Burch, A. Schultz, B. Burger, and D. Weiskopf. Visualization of time series data with spatial context: communicating the energy production of power plants. In *Proceedings of the 10th International Symposium on Visual Information Communication and Interaction*, VINCI ’17, pages 37–44. ACM, 2017.

[13] B. Höferlin, R. Netzel, M. Höferlin, D. Weiskopf, and G. Heidemann. Interactive learning of ad-hoc classifiers for video visual analytics. In *IEEE Conference on Visual Analytics Science and Technology (VAST), 2012*, pages 23–32, 2012.

Bibliography

[14] S. Starke. Texturadvektion höherer Ordnung. Abschlussarbeit (Diplom), Universität Stuttgart, 2011.

[15] A. Haug. Simultaneous contrast compensation of perceived directions. Thesis (Diplom), University of Stuttgart, 2017.

[16] S. Balakrishnan. Comparative eye tracking study of annotated map visualizations. Bachelor thesis, University of Stuttgart, 2015.

[17] B. Ohlhausen. Evaluation of metro map designs by an eye tracking study. Thesis (Diplom), Universtiy of Stuttgart, 2014.

[18] C. W. Oyster. *The Human Eye: Structure and Function*. Oxford University Press, 1999.

[19] D. Atchison and G. Smith. *Optics of the Human Eye*. Elsevier, 2000.

[20] M. D. Fairchild. *Color Appearance Models*. Wiley, 2013.

[21] T. W. Vanderah and D. J. Gould. *Nolte's The Human Brain: An Introduction to its Functional Anatomy*. Elsevier Ltd, Oxford, 7 edition, 2015.

[22] A. P. Schachat, C. P. Wilkinson, D. R. Hinton, S. R. Sadda, and P. Wiedemann. *Ryan's Retina*. Elsevier, 2017.

[23] M. Land, N. Mennie, and J. Rusted. The roles of vision and eye movements in the control of activities of daily living. *Perception*, 28(11):1311–1328, 1999.

[24] M. A. Just and P. A. Carpenter. Eye fixations and cognitive processes. *Cognitive Psychology*, 8(4):441–480, 1976.

[25] R. Miikkulainen, J. A. Bednar, Y. Choe, and J. Sirosh. *Computational Maps in the Visual Cortex*. Springer-Verlag, New York, 2005.

[26] L. A. Remington. *Clinical Anatomy and Physiology of the Visual System*. Butterworth-Heinemann, 2012.

[27] S. Molotchnikoff and J. Rouat, editors. *Visual Cortex - Current Status and Perspectives*. 2012.

[28] R. Carter. *The Brain*. Dorling Kindersley Ltd, 2014.

[29] A. L. Yarbus. *Eye Movements and Vision*. New York: Plenum Press, 1967.

[30] D. W. Hansen and Q. Ji. In the eye of the beholder: a survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):478–500, 2010.

[31] A. T. Duchowski. *Eye Tracking Methodology: Theory and Practice*. Springer-Verlag, New York, 2007.

[32] N. Flad, T. Fomina, H. H. Buelthoff, and L. L. Chuang. Unsupervised clustering of EOG as a viable substitute for optical eye tracking. In M. Burch, L. Chuang, B. Fisher, A. Schmidt, and D. Weiskopf, editors, *Eye Tracking and Visualization*, pages 151–167, Cham, 2017. Springer International Publishing.

[33] L. R. Young and D. Sheena. Survey of eye movement recording methods. *Behavior Research Methods & Instrumentation*, 7(5):397–429, 1975.

[34] D. D. Salvucci and J. H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00, pages 71–78, 2000.

[35] J. San Agustin. Off-the-shelf gaze interaction. PhD thesis, IT-Universitetet i Kobenhavn, Denmark, 2010.

[36] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer. *Eye Tracking: A Comprehensive Guide to Methods and Measures*. Oxford University Press, 2011.

[37] K. Krejtz, A. Coltekin, A. Duchowski, and A. Niedzielska. Using coefficient k to distinguish ambient/focal visual attention during map viewing. *Journal of Eye Movement Research*, 10(2), 2017.

[38] V. Veeraraghavan and S. Shetgovekar. *Textbook of parametric and nonparametric statistics*. SAGE Publications, 2016.

[39] D. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. Taylor & Francis, fifth edition edition, 2011.

[40] G. Cumming. *Understanding the New Statistics: Effect Sizes, Confidence Intervals, and Meta-analysis*. Routledge, 2011.

[41] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl. State-of-the-art of visualization for eye tracking data. In R. Borgo and R. Maciejewski and I. Viola, editor, *EuroVis - STARs*, pages 63–82. The Eurographics Association, 2014.

[42] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In *ACM SIGGRAPH '93*, pages 263–270, 1993.

[43] J. J. van Wijk. Spot noise texture synthesis for data visualization. *Computer Graphics (Proceedings of SIGGRAPH '85)*, 25(4):309–318, 1991.

[44] G.-S. Li, X. Tricoche, and C. Hansen. Physically-based dye advection for flow visualization. *Computer Graphics Forum*, 27(3):727–734, 2008.

[45] D. Weiskopf. Dye advection without the blur: a level-set approach for texture-based visualization of unsteady flow. *Computer Graphics Forum*, 23:479–488, 2004.

[46] F. Neyret. Advected textures. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 147–153, 2003.

[47] M. Y. Hussaini, G. Erlebacher, and B. Jobard. Real-time visualization of unsteady vector fields. In *AIAA Aerospace Sciences Meeting*, 2002.

[48] J. J. van Wijk. Image based flow visualization. *ACM Transactions on Graphics*, 21(3):745–754, 2002.

[49] B. Jobard, G. Erlebacher, and M. Hussaini. Lagrangian-Eulerian advection of noise and dye textures for unsteady flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):211–222, 2002.

[50] D. Weiskopf, R. P. Botchen, and T. Ertl. Interactive visualization of divergence in unsteady flow by level-set dye advection. In *Conference on Simulation and Visualization*, pages 221–232, 2005.

[51] Q. Yu, F. Neyret, E. Bruneton, and N. Holzschuch. Lagrangian texture advection: preserving both spectrum and velocity field. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1612–1623, 2011.

[52] N. Max and B. Becker. Flow visualization using moving textures. In *ICAS/LaRC Symposium on Visualizing Time-Varying Data*, pages 77–87, 1996.

[53] J. Stam. Stable fluids. In *ACM SIGGRAPH '99*, pages 121–128, 1999.

[54] B. Kim, Y. Liu, I. Llamas, and J. Rossignac. Flowfixer: using BFECC for fluid simulation. In *Workshop on Natural Phenomena*, pages 51–56, 2005.

[55] A. Selle, R. Fedkiw, B. Kim, Y. Liu, and J. Rossignac. An unconditionally stable MacCormack method. *Journal of Scientific Computing*, 35(2-3):350–371, 2008.

[56] G. K. Karch, F. Sadlo, D. Weiskopf, C.-D. Munz, and T. Ertl. Visualization of advection-diffusion in unsteady fluid flow. *Computer Graphics Forum*, 31(3):1105–1114, 2012.

[57] E. Hairer, C. Lubich, and G. Wanner. Geometric numerical integration illustrated by the Störmer/Verlet method. *Acta Numerica*, 12:399–450, 2003.

[58] J. Butcher and G. Wanner. Runge-kutta methods: some historical notes. *Applied Numerical Mathematics*, 22:113–151, 1996.

[59] S. Mittelstädt, A. Stoffel, and D. A. Keim. Methods for compensating contrast effects in information visualization. *Computer Graphics Forum*, 33 (3):231–240, 2014.

[60] S. Mittelstädt and D. A. Keim. Efficient contrast effect compensation with personalized perception models. *Computer Graphics Forum*, 34(3):211–220, 2015.

[61] O. Schwartz, T. J. Sejnowski, and P. Dayan. Perceptual organization in the tilt illusion. *Journal of Vision*, 9(4):19–19, 2009.

[62] M. A. Georgeson. Spatial frequency selectivity of a visual tilt illusion. *Nature*, 245:43, 1973.

[63] S. Durant and C. W. G. Clifford. Dynamics of the influence of segmentation cues on orientation perception. *Vision Research*, 46(18):2934–2940, 2006.

[64] H. Wei, Y. Ren, and B.-M. Li. A collaborative decision-making model for orientation detection. *Applied Soft Computing*, 13(1):302–314, 2013.

[65] B. Franceschiello, A. Sarti, and G. Citti. A neuro-mathematical model for geometrical optical illusions. *Journal of Mathematical Imaging and Vision*, 60(1):94–108, 2018.

[66] D. Weiskopf. Iterative twofold line integral convolution for texture-based vector field visualization. In T. Möller, B. Hamann, and R. Russell, editors, *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, pages 191–211. Springer, 2009.

[67] G. M. Karniadakis and R. M. Kirby. *Parallel Scientific Computing in C++ and MPI*. Cambridge University Press, 2003.

[68] R. Bridson and M. Müller-Fischer. Fluid simulation. In *ACM SIGGRAPH 2007 Course Notes*, pages 1–81, 2007.

[69] T. F. Dupont and Y. Liu. Back and forth error compensation and correction methods for removing errors induced by uneven gradients of the level set function. *Journal of Computational Physics*, 190(1):311–324, 2003.

[70] T. F. Dupont and Y. Liu. Back and forth error compensation and correction methods for semi-lagrangian schemes with application to level set interface computations. *Mathematics of Computation*, 76:647–668, 2007.

[71] B. Jahne. *Practical Handbook on Image Processing for Scientific and Technical Applications, Second Edition*. CRC Press, Inc., 2004.

[72] C. W. Clifford, J. Pearson, J. D. Forte, and B. Spehar. Colour and luminance selectivity of spatial and temporal interactions in orientation perception. *Vision Research*, 43(27):2885–2893, 2003.

[73] W. Lovegrove and R. Over. Colour selectivity in orientation masking and aftereffect. *Vision Research*, 13(5):895–901, 1973.

[74] C. W. G. Clifford, B. Spehar, S. G. Solomon, P. R. Martin, and Z. Qasim. Interactions between color and luminance in the perception of orientation. *Journal of Vision*, 3(2):106–115, 2003.

[75] P. Vaidyanathan, E. Prud'hommeaux, J. B. Pelz, C. O. Alm, and A. R. Haake. Fusing eye movements and observer narratives for expert-driven image-region annotations. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ETRA '16, pages 27–34. ACM, 2016.

[76] S. Benayoun, H. Bernard, P. Bertolino, P. Bouthemy, M. Gelgon, R. Mohr, C. Schmid, and F. Spindler. Structuring video documents for advanced interfaces. ACM Multimedia Conference, Demo session, 1998.

[77] P. Bertolino. Sensarea: an authoring tool to create accurate clickable videos. In *Proceedings of 10th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–4, 2012.

[78] L. Zhi and Y. Jie. Interactive video object segmentation: fast seeded region merging approach. *Electronics Letters*, 40(5):302–304, 2004.

[79] D. Ring and A. Kokaram. Feature-cut: video object segmentation through local feature correspondences. In *Proceedings of IEEE 12th International Conference on Computer Vision (ICCV)*, pages 617–624, 2009.

[80] D. Doermann and D. Mihalcik. Tools and techniques for video performance evaluation. In *Proceedings of 15th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 167–170, 2000.

[81] M. Kipp. Anvil: a universal video research tool. In J. Durand, U. Gut, and G. Kristofferson, editors, *Handbook of Corpus Phonology*, pages 420–436. Oxford University Press, 2012.

[82] D. C. Robles, P. Desenne, L. Muellner, and G. Nagy. Open video annotation project, 2015, http://www.openvideoannotation.org.

[83] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, 101: 1–21, 2012.

[84] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: a large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[85] S. N. H. Haji Mirza, M. Proulx, and E. Izquierdo. Gaze movement inference for user adapted image annotation and retrieval. In *Proceedings of the 2011 ACM Workshop on Social and Behavioural Networked Media Access (SBNMA)*, pages 27–32, 2011.

[86] A. Fathi, M. F. Balcan, X. Ren, and J. M. Rehg. Combining self training and active learning for video segmentation. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 78.1–78.11, 2011.

[87] J. H. Goldberg and J. I. Helfman. Visual scanpath representation. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA ’10, pages 203–210, 2010.

[88] M. Raschke, D. Herr, T. Blascheck, T. Ertl, M. Burch, S. Willmann, and M. Schrauf. A visual approach for scan path comparison. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA ’14, pages 135–142, 2014.

[89] A. T. Duchowski, M. M. Price, M. Meyer, and P. Orero. Aggregate gaze visualization with real-time heatmaps. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications*, pages 13–20, 2012.

[90] B. Tversky, J. B. Morrison, and M. Betrancourt. Animation: can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262, 2002.

[91] G. G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. T. Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, 2008.

[92] K. Kurzhals and D. Weiskopf. Space-time visual analytics of eye-tracking data for dynamic stimuli. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2129–2138, 2013.

[93] G. Andrienko, N. Andrienko, M. Burch, and D. Weiskopf. Visual analytics methodology for eye movement studies. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2889–2898, 2012.

[94] H. Y. Tsang, M. Tory, and C. Swindells. eSeeTrack – Visualizing sequential fixation patterns. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):953–962, 2010.

[95] M. Burch, A. Kull, and D. Weiskopf. AOI rivers for visualizing dynamic eye gaze frequencies. *Computer Graphics Forum*, 32(3):281–290, 2013.

[96] T. Blascheck, K. Kurzhals, M. Raschke, S. Strohmaier, D. Weiskopf, and T. Ertl. AOI hierarchies for visual exploration of fixation sequences. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications*, pages 111–118, 2016.

[97] M. Raschke, X. Chen, and T. Ertl. Parallel scan-path visualization. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications*, pages 165–168, 2012.

[98] J. H. Goldberg, M. J. Stimson, M. Lewenstein, N. Scott, and A. M. Wichansky. Eye tracking in web search tasks: design implications. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications*, pages 51–58, 2002.

[99] R. Bednarik and M. Tukiainen. Temporal eye-tracking data: evolution of debugging strategies with multiple representations. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications*, pages 99–102, 2008.

[100] M. Bader. *Space-Filling Curves: An Introduction with Applications in Scientific Computing*. Springer, Heidelberg, New York, Dordrecht, London, 2012.

[101] H. Sagan and J. Holbrook. *Space-Filling Curves*. Springer, New York, 1994.

[102] M. Wattenberg. A note on space-filling visualizations and space-filling curves. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 181–186, 2005.

[103] C. K. Chen, C. Wang, K. L. Ma, and A. T. Wittenberg. Static correlation visualization for large time-varying volume data. In *Proceedings of the IEEE Pacific Visualization Symposium*, pages 27–34, 2011.

[104] S. Anders. Visualization of genomic data with the hilbert curve. *Bioinformatics*, 25:1231–1235, 2009.

[105] D. A. Keim. Designing pixel-oriented visualization techniques: theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, 2000.

[106] G. Paliouras, C. Papatheodorou, V. Karkaletsis, and C. D. Spyropoulos. Discovering user communities on the internet using unsupervised machine learning techniques. *Interacting with Computers*, 14(6):761–791, 2002.

[107] H.-C. Chen and A. L. Chen. A music recommendation system based on music data grouping and user interests. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 231–238. ACM, 2001.

[108] A. Vernone, P. Berchialla, and G. Pescarmona. Human protein cluster analysis using amino acid frequencies. *PloS one*, 8(4):e60220, 2013.

[109] M. Burch, N. Konevtsova, J. Heinrich, M. Hoeferlin, and D. Weiskopf. Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2440–2448, 2011.

[110] K. Kurzhals, F. Heimerl, and D. Weiskopf. ISeeCube: visual analysis of gaze data for video. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA)*, pages 43–50. ACM, 2014.

[111] X. Li, A. Çöltekin, and M.-J. Kraak. Visual exploration of eye movement data using the space-time-cube. In *International Conference on Geographic Information Science*, pages 295–309. Springer, 2010.

[112] N. C. Anderson, F. Anderson, A. Kingstone, and W. F. Bischof. A comparison of scanpath comparison methods. *Behavior Research Methods*, 47(4): 1377–1392, Dec 2015.

[113] J. M. West, A. R. Haake, E. P. Rozanski, and K. S. Karn. eyepatterns: software for identifying patterns and similarities across fixation sequences. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, pages 149–154. ACM, 2006.

[114] M. Burch, A. Kumar, K. Mueller, and D. Weiskopf. Color bands: visualizing dynamic eye movement patterns. In *2016 IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)*, pages 40–44, 2016.

[115] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz. Analysis of the clustering properties of the Hilbert space-filling curve. *IEEE Transactions on knowledge and data engineering*, 13(1):124–141, 2001.

[116] M. Tory, C. Swindells, and R. Dreezer. Comparing dot and landscape spatializations for visual memory differences. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1033–1040, 2009.

[117] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 110–119, 2000.

[118] K. Kurzhals, C. F. Bopp, J. Bässler, F. Ebinger, and D. Weiskopf. Benchmark data for evaluating visualization and analysis techniques for eye tracking for video stimuli. In *Proceedings of BELIV'14: Beyond time and errors: novel evaluation methods for information visualization*, pages 54–60, 2014.

[119] T. Mihalisin, J. Timlin, and J. Schwegler. Visualizing multivariate functions, data, and distributions. *IEEE Computer Graphics and Applications*, 11(3): 28–35, 1991.

[120] J. LeBlanc, M. O. Ward, and N. Wittels. Exploring n-dimensional databases. In *Proceedings of the 1st Conference on Visualization '90*, VIS '90, pages 230–237, 1990.

[121] C. Beshers and S. Feiner. Autovisual: rule-based design of interactive multivariate visualizations. *IEEE Computer Graphics and Applications*, 13 (4):41–49, 1993.

[122] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., 2004.

[123] D. Holten and J. J. van Wijk. A user study on visualizing directed edges in graphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2299–2308, 2009.

[124] D. Holten, P. Isenberg, J. van Wijk, and J. Fekete. An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. In *Proceedings of the IEEE Pacific Visualization Symposium*, pages 195–202, 2011.

[125] G. di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.

[126] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[127] C. G. Healey and J. T. Enns. Large datasets at a glance: combining textures and colors in scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):145–167, 1999.

[128] A. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.

[129] U. Engelke, J. Vuong, and J. Heinrich. Visual performance in multidimensional data characterisation with scatterplots and parallel coordinates. In *Proceedings of IS&T Human Vision and Electronic Imaging (HVEI)*, Feb. 2016.

[130] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 111–117, 2005.

[131] R. Nathan and L. Giuggioli. A milestone for movement ecology research. *Movement Ecology*, 1:1–3, 2013.

[132] J. Shamoun-Baranes, E. van Loon, R. Purves, B. Speckmann, D. Weiskopf, and C. Camphuysen. Analysis and visualization of animal movement. *Biology Letters*, 8(1):6–9, 2012.

[133] M. Burch, C. Vehlow, N. Konevtsova, and D. Weiskopf. Evaluating partially drawn links for directed graph edges. In *Proceedings of the International Conference on Graph Drawing*, pages 226–237, 2012.

[134] T. Bruckdorfer, S. Cornelsen, C. Gutwenger, M. Kaufmann, F. Montecchiani, M. Nöllenburg, and A. Wolff. Progress on partial edge drawings. In *Proceedings of the International Conference on Graph Drawing*, pages 67–78, 2013.

[135] R. Jianu, A. Rusu, A. Fabian, and D. Laidlaw. A coloring solution to the edge crossing problem. In *Proceedings of the Conference on Information Visualisation*, pages 691–696, 2009.

[136] A. Rusu, A. Fabian, R. Jianu, and A. Rusu. Using the Gestalt principle of closure to alleviate the edge crossing problem in graph drawings. In

Proceedings of the Conference on Information Visualisation, pages 488–493, 2011.

[137] G. Andrienko, N. Andrienko, P. Bak, D. Keim, and S. Wrobel. *Visual Analytics of Movement*. Springer, 2013.

[138] M. Höferlin, B. Höferlin, G. Heidemann, and D. Weiskopf. Interactive schematic summaries for faceted exploration of surveillance video. *IEEE Transactions on Multimedia*, 15(4):908–920, 2013.

[139] H. Janetzko, D. Jäckle, O. Deussen, and D. Keim. Visual abstraction of complex motion patterns. In *Proceedings of S&T/SPIE Electronic Imaging*, volume 9017, pages 90170J–90170J–12, 2013.

[140] N. Willems, H. van de Wetering, and J. J. van Wijk. Visualization of vessel movements. In *Proceedings of the 11th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'09, pages 959–966, 2009.

[141] R. Scheepens, N. Willems, H. van de Wetering, and J. van Wijk. Interactive visualization of multivariate trajectory data with density maps. In *Proceedings of the IEEE Pacific Visualization Symposium*, pages 147–154, 2011.

[142] M. Burch, C. Vehlow, F. Beck, S. Diehl, and D. Weiskopf. Parallel edge splatting for scalable dynamic graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2344–2353, 2011.

[143] R. Van Liere and W. de Leeuw. GraphSplatting: visualizing graphs as continuous fields. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):206–212, 2003.

[144] D. Pilar and C. Ware. Representing flow patterns by using streamlines with glyphs. *IEEE Transactions on Visualization and Computer Graphics*, 19 (8):1331–1341, 2013.

[145] R. S. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F. H. Post, and D. Weiskopf. The state of the art in glow visualization: dense and texture-based techniques. *Computer Graphics Forum*, 23(2):203–222, 2004.

[146] R. Wegenkittl, E. Gröller, and W. Purgathofer. Animating flowfields: rendering of oriented line integral convolution. In *Proceedings of Computer Animation*, pages 15–21, 1997.

[147] D. Fowler and C. Ware. Strokes for representing univariate vector field maps. In *Proceedings of Graphics Interface*, pages 249–253, 1989.

[148] J. Goldberg and J. Helfman. Eye tracking for visualization evaluation: reading values on linear versus radial graphs. *Information Visualization*, 10(3):182–195, 2011.

[149] M. Burch, N. Konevtsova, J. Heinrich, M. Höferlin, and D. Weiskopf. Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2440–2448, 2011.

[150] S. Garlandini and S. I. Fabrikant. Evaluating the effectiveness and efficiency of visual variables for geographic information visualization. In *Proceedings of the International Conference on Spatial Information Theory*, pages 195–211, 2009.

[151] E. Halley. An historical account of the trade winds, and monsoons, observable in the seas between and near the tropicks, with an attempt to assign the phisical cause of the said winds. *Philosophical Transactions*, 16: 153–168, 1686/1692.

[152] J. Bertin. *Semiology of Graphics*. University of Wisconsin Press, 1983.

[153] C. Ware. Toward a perceptual theory of flow visualization. *IEEE Computer Graphics and Applications*, 28(2):6–11, 2008.

[154] M. Ghoniem, J. Fekete, and P. Castagliola. A comparison of the readability of graphs using node-link and matrix-based representations. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 17–24, 2004.

[155] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. Task taxonomy for graph visualization. In *Proceedings of BELIV*, pages 1–5, 2006.

[156] D. Somayeh, W. Robert, and A.-K. Lautenschütz. Towards a taxonomy of movement patterns. *Information Visualization*, 7(3-4):240–252, 2008.

[157] C. G. Healey and J. T. Enns. Attention and visual memory in visualization and computer graphics. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1170–1188, 2012.

[158] L. Lischke, S. Mayer, K. Wolf, N. Henze, A. Schmidt, S. Leifert, and H. Reiterer. Using space: effect of display size on users' search performance. In *Proceedings of the ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA)*, pages 1845–1850, 2015.

[159] J. Müller, R. Rädle, H.-C. Jetter, and H. Reiterer. An experimental comparison of vertical and horizontal dynamic peephole navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1523–1526, 2015.

[160] C. Ware. *Visual Thinking for Design*. Morgan Kaufmann, 2008.

[161] G. S. Srivastava. *An Introduction to Geoinformatics*. McGraw-Hill Education, 2014.

[162] M. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff. Boundary labeling: models and efficient algorithms for rectangular maps. *Computational Geometry*, 36(3):215 – 236, 2007.

[163] R. Rosenholtz, Y. Li, J. Mansfield, and Z. Jin. Feature congestion: a measure of display clutter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 761–770, 2005.

[164] P. Baudisch and R. Rosenholtz. Halo: a technique for visualizing off-screen objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 481–488, 2003.

[165] P. Kindermann, B. Niedermann, I. Rutter, M. Schaefer, A. Schulz, and A. Wolff. Multi-sided boundary labeling. *Algorithmica*, 74:1–34, 2015.

[166] E. Tufte. *Beautiful Evidence*. Graphics Press, 2006.

[167] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 265–272, 1995.

[168] M. Burch and S. Diehl. TimeRadarTrees: visualizing dynamic compound digraphs. *Computer Graphics Forum*, 27(3):823–830, 2008.

[169] J. Wood, J. Dykes, and A. Slingsby. Visualisation of origins, destinations and flows with OD maps. *The Cartographic Journal*, 47(2):117–129, 2010.

[170] A. H. Robinson. *The Look of Maps: An Examination of Cartographic Design*. Esri Press, 2010, 1st reprint.

[171] A. H. Robinson, J. L. Morrison, P. C. Muehrcke, A. J. Kimerling, and S. C. Guptill. *Elements of Cartography*. Hardcover, 3rd edition, 1995.

[172] J. A. Tyner. *Principles of Map Design*. Guilford Press, 2010.

[173] H.-Y. Wu, S. Takahashi, C.-C. Lin, and H.-C. Yen. Voronoi-based label placement for metro maps. In *Proceedings of the International Conference on Information Visualisation (IV)*, pages 96–101, 2013.

[174] H. A. do Nascimento and P. Eades. User hints for map labeling. *Journal of Visual Languages & Computing*, 19(1):39–74, 2008.

[175] J. Dykes, J. Wood, and A. Slingsby. Rethinking map legends with visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6): 890–899, 2010.

[176] M. Birsak, P. Musialski, P. Wonka, and M. Wimmer. Automatic generation of tourist brochures. *Computer Graphics Forum*, 33(2):449–458, 2014.

[177] H.-Y. Wu, S.-H. Poon, S. Takahashi, M. Arikawa, C.-C. Lin, and H.-C. Yen. Designing and annotating metro maps with loop lines. In *Proceedings of the International Conference on Information Visualisation (IV)*, pages 9–14, 2015.

[178] I. Golebiowska. Legend layouts for thematic maps: a case study integrating usability metrics with the thinking aloud method. *The Cartographic Journal*, 52(1):28–40, 2015.

[179] C. Ebert. Correspondence visualization techniques for analyzing and evaluating software measures. *IEEE Transactions on Software Engineering*, 18(11):1029–1034, 1992.

[180] M. Maciel, B. Meiguins, R. Loureno, A. Meiguins, and P. Godinho. The impact of multiple coordinated views on the visual data exploration and analysis. In *Proceedings of the International Conference on Information Visualisation (IV)*, pages 113–119, 2008.

[181] H. Bouma and J. Andriessen. Perceived orientation of isolated line segments. *Vision Research*, 8(5):493 – 507, 1968.

[182] R. F. Hess, S. C. Dakin, and N. Kapoor. The foveal ‘crowding’ effect: physics or physiology? *Vision Research*, 40(4):365–370, 2000.

[183] J. M. Loomis, J. A. Da Silva, J. W. Philbeck, and S. S. Fukusima. Visual perception of location and distance. *Current Directions in Psychological Science*, 5(3):72–77, 1996.

[184] D. McCready. On size, distance, and visual angle perception. *Perception & Psychophysics*, 37(4):323–334, 1985.

[185] M. Ovenden, L. T. Museum, and C. Transport. *Metro Maps of the World*. Capital Transport, 2005.

[186] F. Wagner, A. Wolff, V. Kapoor, and T. Strijk. Three rules suffice for good label placement. *Algorithmica*, 30(2):334–349, 2001.

[187] K. Garland and H. Beck. *Mr Beck's Underground Map*. Capital Transport, 1994.

[188] A. Dow. *Telling the Passenger where to Get Off: George Dow and the Evolution of the Railway Diagrammatic Map*. Capital Transport Pub., 2005.

[189] M. A. C. Horne. Information design aspects of the London underground map, 2012. [Online]. Available: `http://www.metadyne.co.uk/UndMap.html`.

[190] P. Kiefer, I. Giannopoulos, D. Kremer, C. Schlieder, and M. Raubal. Starting to get bored: an outdoor eye tracking study of tourists exploring a city panorama. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA)*, pages 315–318. ACM, 2014.

[191] A. Brychtova and A. Coltekin. An empirical user study for measuring the influence of colour distance and font size in map reading using eye tracking. *Cartographic Journal*, 51(4):1–11, 2014.

[192] K. Krejtz, A. T. Duchowski, and A. Çöltekin. High-level gaze metrics from map viewing - charting ambient/focal visual attention. In *Proceedings of the 2nd International Workshop on Eye Tracking for Spatial Research*, pages 37–41. CEUR-WS.org, 2014.

[193] P. Kiefer, I. Giannopoulos, and M. Raubal. Using eye movements to recognize activities on cartographic maps. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 488–491. ACM, 2013.

[194] K. Ooms, A. Coltekin, P. D. Maeyer, L. Dupont, S. I. Fabrikant, A. Incoul, M. Kuhn, H. Slabbinck, P. Vansteenkiste, and L. V. der Haegen. Combining user logging with eye tracking for interactive and dynamic applications. *Behavior Research Methods*, 47(4):977–993, 2015.

[195] A. Çöltekin, S. Fabrikant, and M. Lacayo. Exploring the efficiency of users' visual analytics strategies based on sequence analysis of eye movement recordings. *International Journal of Geographical Information Science*, 24(10): 1559–1575, 2010.

[196] W. Huang, P. Eades, and S.-H. Hong. A graph reading behavior: geodesic-path tendency. In *Proceedings of the IEEE Pacific Visualization Symposium*, pages 137–144, 2009.

[197] M. Burch, G. Andrienko, N. Andrienko, M. Höferlin, M. Raschke, and D. Weiskopf. Visual task solution strategies in tree diagrams. In *Proceedings of the IEEE Pacific Visualization Symposium*, pages 169–176, 2013.

[198] M. Burch, H. Schmauder, M. Raschke, and D. Weiskopf. Saccade plots. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '14, pages 307–310, New York, NY, USA, 2014. ACM.

[199] M. Burch, M. Raschke, T. Blascheck, K. Kurzhals, and D. Weiskopf. How do people read metro maps? An eye tracking study. In *Proceedings of 1st International Workshop on Schematic Mapping*, 2014.

[200] Y. Yesilada, S. Harper, and S. Eraslan. Experiential transcoding: an eye-tracking approach. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, W4A '13, pages 30:1–30:4. ACM, 2013.

[201] P. Hejmady and N. H. Narayanan. Visual attention patterns during program debugging with an IDE. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA)*, ETRA '12, pages 197–200. ACM, 2012.

[202] P. Kiefer and I. Giannopoulos. Gaze map matching: mapping eye tracking data to geographic vector features. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pages 359–368. ACM, 2012.

[203] W. S. Cleveland and R. McGill. Graphical perception: the visual decoding of quantitative information on graphical displays of data. *Journal of the Royal Statistical Society. Series A (General)*, 150(3):192–229, 1987.

[204] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1 (4):69–91, 1985.

[205] J. Li, J.-B. Martens, and J. J. van Wijk. Judging correlation from scatterplots and parallel coordinate plots. *Information Visualization*, 9(1):13–30, 2010.

[206] L. Harrison, F. Yang, S. Franconeri, and R. Chang. Ranking visualizations of correlation using Weber's law. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1943–1952, 2014.

[207] D. Holten and J. J. V. Wijk. Evaluation of cluster identification performance for different PCP variants. *Computer Graphics Forum*, 29(3):793–802, 2010.

[208] X. Kuang, H. Zhang, S. Zhao, and M. J. McGuffin. Tracing tuples across dimensions: a comparison of scatterplots and parallel coordinate plots. *Computer Graphics Forum*, 31(3):1365–1374, 2012.

[209] J. H. T. Claessen and J. J. van Wijk. Flexible linked axes for multivariate data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2310–2316, 2011.

[210] J. Hartigan. Printer graphics for clustering. *Journal of Statistical Computation and Simulation*, 4(3):187–213, 1975.

[211] H. Qu, W.-Y. Chan, A. Xu, K.-L. Chung, K.-H. Lau, and P. Guo. Visual analysis of the air pollution problem in Hong Kong. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1408–1415, 2007.

[212] C. Viau, M. J. McGuffin, Y. Chiricota, and I. Jurisica. The FlowVizMenu and parallel scatterplot matrix: hybrid multidimensional visualizations for network exploration. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1100–1108, 2010.

[213] J. Heinrich and D. Weiskopf. State of the art of parallel coordinates. In *Eurographics 2013 - State of the Art Reports*, pages 95–116, 2013.

[214] J. Johansson and C. Forsell. Evaluation of parallel coordinates: overview, categorization and guidelines for future research. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):579–588, 2016.

[215] R. Kanjanabose, A. Abdul-Rahman, and M. Chen. A multi-task comparative study on scatter plots and parallel coordinates plots. *Computer Graphics Forum*, 34(3):261–270, 2015.

[216] M. Henley, M. Hagen, and R. D. Bergeron. Evaluating two visualization techniques for genome comparison. In *Proceedings of the International Conference on Information Visualization (IV)*, pages 551–558, 2007.

[217] H. Siirtola, T. Laivo, T. Heimonen, and K. J. Räihä. Visual perception of parallel coordinate visualizations. In *Proceedings of the International Conference on Information Visualization (IV)*, pages 3–9, 2009.

[218] D. Simkin and R. Hastie. An information-processing analysis of graph perception. *Journal of the American Statistical Association*, 82(398):454–465, June 1987.

[219] M. Harrower and C. A. Brewer. ColorBrewer.org: an online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[220] L. F. Lu, M. L. Huang, and T. H. Huang. A new axes re-ordering method in parallel coordinates visualization. In *2012 11th International Conference on Machine Learning and Applications*, volume 2, pages 252–257, 2012.

[221] G. Albuquerque, M. Eisemann, D. J. Lehmann, H. Theisel, and M. Magnor. Improving the visual analysis of high-dimensional datasets using quality measures. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pages 19–26, 2010.

[222] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnork, and D. Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pages 59–66, 2009.

[223] A. Dasgupta and R. Kosara. Pargnostics: screen-space metrics for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 16 (6):1017–1026, 2010.